D. Watt
Il Logo per il
Commodore 64
McGraw Hill

Il Logo per il Commodore 64

D. Watt

Il Logo per il Commodore 64

McGRAW-HILL Libri Italia srl

Milano · New York · St Louis · San Francisco · Amburgo · Auckland
Bogotá · Città del Guatemala · Città del Messico · Johannesburg
Lisbona · Londra · Madrid · Montreal · Nuova Delhi · Panama · Parigi
San Juan · San Paolo · Singapore · Sydney · Tokyo · Toronto

Un libro di BYTE

Titolo originale: *Learning With Commodore Logo*

Realizzazione editoriale: EDIGEO srl, via del Lauro 3, 20121 Milano
Traduzione: Mauro Formaggio
Copertina: Marco Mazzucato
Composizione e stampa: Litovelox, Trento

ISBN 88-386-0042-2

1ª edizione aprile 1986

Indice

Ringraziamenti

Molte sono le persone a cui sono riconoscente per le idee che mi hanno dato e per il lavoro che hanno fatto durante la stesura di questo libro.
Il mio primo pensiero va a Molly Watt, con la quale ho discusso a fondo ogni idea e che è sempre riuscita a migliorare quanto le andavo presentando; molte idee, poi, sono state sperimentate proprio nei seminari da lei tenuti al Lesley College e al Keene State College (dedicati agli insegnanti) e al Computer Camps International (dedicati ai ragazzi). Come se ciò non bastasse, Molly ha anche dovuto provare l'esperienza, per nulla da invidiare, di vivere con me durante i mesi veramente defatiganti in cui questo libro è diventato realtà: di ciò le sono profondamente grato, come del resto già sa.
Il libro si avvale dell'esperienza che ho fatto, a partire dalla fine degli anni sessanta, partecipando all'Elementary Science Study, un progetto di sviluppo scolastico finanziato con fondi del governo federale, grazie al quale migliaia di allievi e di insegnanti delle scuole elementari hanno potuto conoscere il mondo della scienza. Fu in quell'occasione che maturai la mia metodologia educativa, confrontando le mie idee con quelle dei colleghi nel progetto, e mettendole poi in pratica nei sette anni trascorsi alla scuola elementare di Brookline, nel Massachusetts.
Quando poi entrai a far parte del Logo Group al MIT (*Massachusetts Institute of Technology*) compresi che veramente le idee educative maturate negli anni sessanta e settanta potevano tradursi in pratica reale e in esperienza autentica per migliaia, anzi milioni di persone, e questo grazie alla microelettronica, al linguaggio Logo e alle idee ispiratrici di Seymour Papert.
Il Logo non è infatti solo un linguaggio di programmazione, ma è ormai

divenuto un ambiente culturale, per merito di molti e soprattutto, in modo speciale, di Seymour Papert. Egli fu infatti il massimo teorico del Logo e anche l'artefice principale della sua applicazione pratica. Formò i gruppi di ricerca e ottenne i fondi per l'hardware, il software, per lo sviluppo della metodologia educativa e l'approfondimento della filosofia dell'apprendimento su cui si basa il Logo. Lavorare con lui fu per me un'esperienza fondamentale.
Molti altri parteciparono al gruppo di sviluppo del Logo, nei cinque anni del mio apprendistato. Ricordo specialmente coloro che lavorarono con me ai due progetti educativi di Brookline, fra il 1977 e il 1981, e che hanno il merito di molte fra le idee che concorrono a questo libro: Hal Abelson, Jeanne Bamberger, Andy diSessa, Greg Gargarian, Ellen Hildreth, Danny Hillis, Bob Lawler, Margaret Minsky, Seymour Papert, Sylvia Weir e Ursula Wolz. Altri che contribuirono molto alla formazione dell'ambiente culturale del Logo al MIT furono Howard Austin, John Berlow, Paul Goldenberg, Ginny Grammer, Marvin Minsky, Cynthia Solomon e Jose Valente.
Vorrei ringraziare anche tutti gli allievi della Lincoln School di Brookline (diverse centinaia di ragazzi, fra i 9 e i 14 anni): gran parte del libro si basa sulle loro idee ed esperienze. Fu per merito del loro entusiasmo e della loro applicazione che il Logo potè uscire dalla torre d'avorio della teoria dell'apprendimento e diventare una realtà della mia vita. Non voglio dimenticare qui gli insegnanti e gli amministratori che tennero in vita il centro Logo alla Lincoln School per cinque anni: Joan Aronson, Gerry Cote, Lisa Hirsch, Bob Lewis, Mary Parkins, Florence Regolino, Ellie Shacter, Robin Welch e il sovrintendente scolastico della città di Brookline, Bob Sperber, che credette fra i primi alle possibilità di impiego pratico del Logo, a dispetto dell'incredulità dei più.
Alcuni hanno contribuito a questo libro in modo diretto, con idee o applicazioni pratiche, da me riprese. Molti dei progetti grafici dei Capitoli 5, 6 e 7 sono dovuti a Hellen Hildreth; altri, specie quelli del Capitolo 8, sono tratti dal libro *Turtle Geometry* di Hal Abelson e Andy diSessa, pubblicato dalla MIT Press. Paul Goldenberg mi presentò per primo il programma SCHIZZO, usato nei Capitoli 3 e 11, anche se non era il solo ad impiegarlo, al MIT. Per il TIRO.A.SEGNO, invece, il gioco presentato nei Capitoli 3 e 10, sono debitore a Bob Lawler, che lo realizzò per i suoi bambini. L'idea della tartaruga è di Danny Hillis, sviluppata poi in molti giochi dagli allievi della Lincoln School. Le procedure POETA sono tratte da un programma BASIC pubblicato sulla rivista *Classroom Computer News* di Lloyd Prentice; l'idea è comunque di molto anteriore, e risale ai primordi stessi del Logo, al MIT, quando ancora nessuno aveva pensato alla tartaruga.

Prima di incominciare

Questo libro intende aiutare il lettore ad usare un computer; per questo comincia con alcuni commenti volti a facilitarne la lettura. Può sembrare strano che un libro inizi con osservazioni che non riguardano il contenuto, ma la lettura del libro stesso. Chi vuole partire subito può passare al Capitolo 1, chi vuole invece avere un'idea generale di cosa si accinge ad intraprendere, troverà qui alcune indicazioni utili.

A chi si rivolge questo libro

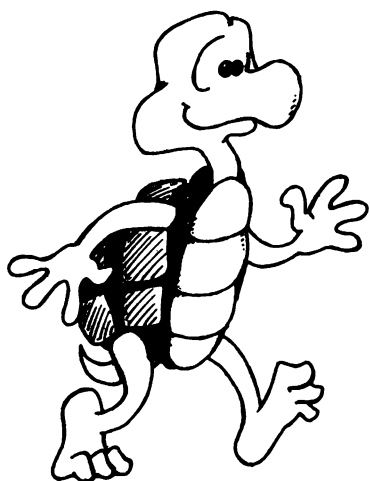
Il lettore può essere chiunque desideri imparare il Logo, un linguaggio che rende facile e persino divertente la programmazione del computer. Il Logo è quello che ci vuole per i ragazzi, perché rende facili i primi passi e non è mai complicato, anche quando si tratta di fare cose davvero entusiasmanti. Il Logo è anche adatto agli adulti, che imparandolo sono in grado di far eseguire al computer compiti impegnativi, niente affatto semplici con altri linguaggi, in un modo sorprendentemente naturale. Insomma, il Logo è per tutti.
Il libro andrebbe usato da piccoli e grandi insieme, in modo che si aiutino a vicenda. I più piccoli, sui 4 o 5 anni, possono limitarsi al Capitolo 3, seguendone le semplici attività con l'aiuto dell'insegnante, del papà o della mamma, o di un amico più grande. I ragazzi di 10 o 11 anni sono in grado di leggere quasi da soli i Capitoli da 1 a 6, e quelli appena più grandi se la dovrebbero cavare anche con il resto del libro, con un aiuto

limitato. I grandi, in genere, non dovrebbero avere difficoltà a leggere il libro, ma si divertiranno molto di più se lo faranno insieme ai piccoli, a meno che non abbiano conservato intatto (e ciò capita a pochi) lo spirito d'avventura e di gioco della loro gioventù.
Gli insegnanti e i genitori possono leggere il libro e presentarne alcune parti ai piccoli, senza che questi leggano l'intero testo: suggerisco però ai grandi di mettersi sempre nei panni di chi impara e di provare il maggior numero possibile di attività, di costruzioni e di esplorazioni, in modo da rendersi ben conto di cosa proveranno i piccoli allievi.
Veramente, per me c'è un solo modo valido per aiutare qualcuno ad imparare il Logo: mi siedo con lui davanti al computer, guardo come lavora, faccio commenti, rispondo alle sue domande e qualche volta gliene pongo. Faccio soprattutto attenzione a notare le difficoltà di chi sta imparando, e anche cosa gli piace davvero o gli riesce particolarmente bene. Quando aiuto qualcuno ad imparare, mi regolo anche secondo la mia esperienza di insegnante, cercando di adattarla alla persona che al momento è con me. Addirittura certe volte è meglio che mi allontani per un certo tempo, lasciando che l'allievo proceda da solo. Altre volte invece partecipo attivamente, indicando ogni nuovo passo e risolvendo magari insieme un problema che risulta difficile per ambedue. Molte volte, aiutando qualcuno o semplicemente osservandolo, imparo io stesso qualcosa.
Quando decisi di scrivere questa guida al Logo e al suo insegnamento, mi rendevo ben conto che la difficoltà maggiore sarebbe derivata dall'esposizione stessa, che necessariamente comporta la scelta dei termini e dell'ordine in cui sono presentati e ovviamente la rinuncia ai molti altri termini e ordini possibili e pur validi. Vorrei comunque riuscire con questo libro ad indurre la gente ad imparare stando insieme, in modo che possa provare le belle esperienze che hanno reso tanto entusiasmante per me l'imparare e l'insegnare il Logo.

Come usare questo libro

Il libro può essere diviso in tre parti. La prima comprende i Capitoli da 1 a 6 e può essere letta da chiunque, a partire da 10 o 11 anni. La seconda è formata dai Capitoli da 7 a 9 ed è adatta ai ragazzi da 11 a 13 anni, che abbiano già letto la prima parte. La terza parte comprende i Capitoli da 10 a 14 ed è adatta a tutti, a partire da 13 o 14 anni di età o anche ai piccoli, se li aiuta un grande.
La prima parte si rivolge ai principianti e può darsi che qualcuno vi si soffermi a lungo. I Capitoli 1 e 2 presentano la tastiera del computer e la tartaruga, una caratteristica del Logo, una specie di piccolo robot che si sposta sullo schermo tracciando figure.

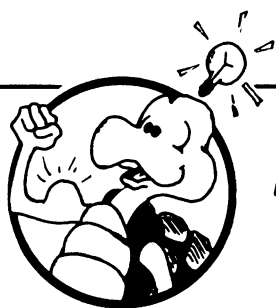
Il Capitolo 3 mostra come svolgere con il Logo alcune attività già programmate: il gioco chiamato TIRO.A.SEGNO che serve a prendere confidenza con l'impiego della tartaruga, e il programma che risponde al nome di SCHIZZO, che permette di eseguire disegni con la tartaruga, pilotandola con i tasti della tastiera.
Il Capitolo 4 insegna il modo di assegnare al computer nuovi comandi, chiamati procedure, come comporre programmi con il Logo (usandone il cosiddetto *screen editor*) e come registrare le nuove procedure sul disco di servizio del Logo. In effetti, il Capitolo 4 è alquanto lungo e anche un po' complicato, tanto che il lettore vi ritornerà ancora, dopo aver letto il resto del libro ed essersi fatta un'idea generale del Logo.
I Capitoli 5 e 6 mostrano come preparare con la tartaruga decine di costruzioni grafiche, realizzate per lo più proprio dai giovani di età compresa fra 9 e 11 anni.
Nella seconda parte sono presentate procedure che comportano l'inserimento di dati da tastiera ed è illustrato il modo in cui il Logo impiega le variabili. Il Capitolo 7 mostra come tracciare con la tartaruga figure con angoli e dimensioni variabili, come ingrandirle o rimpicciolirle, e come impiegare i cosiddetti comandi condizionati, ad esempio per far terminare ciò che il computer sta eseguendo.
Il Capitolo 8 presenta la procedura POLIGONALE, con le sue molte variazioni, e insegna ad eseguire con l'impiego della tartaruga disegni veramente entusiasmanti, basati su una formulazione matematica.
Il Capitolo 9 spiega in che senso e come è possibile dialogare con il computer, e come realizzare giochi e quiz basati su domande e risposte.
La terza parte comprende quattro progetti di sviluppo, che possono iniziare in modo semplice e crescere successivamente in complessità fin che si vuole. Nei Capitoli 10 e 11 il lettore impara a costruire i programmi TIRO.A.SEGNO e SCHIZZO, che sono presentati nel Capitolo 3 come attività già preprogrammate. Il Capitolo 12 mostra poi come far muovere la tartaruga sullo schermo, pilotandola con i tasti della tastiera, e descrive la costruzione di un gioco di azione, in cui la tartaruga rappresenta un'auto da corsa in pista. Il Capitolo 13 considera il modo di far costruire al computer frasi, storie e persino poesie, in base alle parole e agli schemi che gli vengono indicati. Il Capitolo 14 infine spiega il funzionamento interno delle procedure di servizio usate nel libro.
Al termine della terza parte il lettore è finalmente in grado di affrontare in modo autonomo qualunque progetto di sviluppo con il Logo.
Le appendici alla fine del libro raccolgono le informazioni che possono tornare utili al lettore durante la lettura.
L'Appendice A descrive come costruire un proprio disco personalizzato di procedure. In essa sono riportate le liste complete di istruzioni delle procedure di servizio e di molti dei programmi presentati come esempio nel libro.

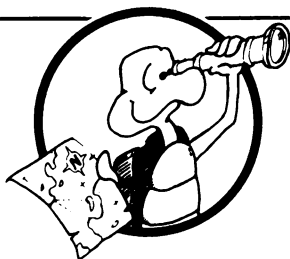
L'Appendice B indica il modo di inizializzare e di ricopiare i dischi di servizio del Logo.
L'Appendice C comprende l'elenco di tutti i comandi del Commodore Logo citati nel libro, e può servire al lettore come guida rapida di riferimento.

La tartaruga

LA TARTARUGA

È un personaggio che diventerà ben presto familiare al lettore. Rappresenta il cursore del Logo, dalla caratteristica forma triangolare e in Logolingua viene chiamato tartaruga, o Turtle.
La tartaruga può anche impersonare, a volte, il lettore stesso.
La nostra tartaruga ci aiuterà anche a leggere questo libro, segnalandone le parti importanti e interessanti.

TRABOCCHETTO

Questo simbolo rappresenta un trabocchetto, cioè una trappola dove è facile cadere, se si è principianti. In effetti, il computer è un po' come un castello da esplorare, con botole, passaggi segreti, ponti levatoi e qualche volta persino fantasmi!
Questo segnale può servire per evitare la trappola, oppure per cavarsi d'impaccio se vi si è cascati.
Non c'è da aver paura delle trappole, e comunque è praticamente impossibile evitarle proprio tutte. E poi, diciamo la verità: così il gioco è molto più emozionante. In fondo, una trappola è anche un'avventura da ricordare e magari raccontare agli amici.
In computerese un errore in un programma è chiamato bug e la caccia di questi errori, per toglierli di mezzo, prende il nome di debugging. I segnali di trabocchetto del libro mettono anche in guardia dai possibili bug (il bug è un avversario insidioso e tenace, sempre temibile e spesso ben mimetizzato: va cacciato con decisione).

IDEA

Questo segnale indica un'idea che può tornare utile al lettore. Una buona trovata può risolvere un problema e può magari servire per raggiungere un traguardo difficile o per superare un ostacolo. Alcune di queste idee valgono anche se non si usa il computer.

ESPLORAZIONE

Il segnale di esplorazione compare quando il lettore, dopo aver visto un argomento nuovo, può esplorare le nuove possibilità del computer, provando da solo. Una vera esplorazione è sempre libera e non può contare su una guida dettagliata, anche perché altrimenti non rimarrebbe molto da scoprire. L'esploratore del computer non trova risposte pronte: le deve scovare nel computer stesso. Certe esplorazioni sono facili, altre sono più impegnative.

SUGGERIMENTO

Questo segnale, che raffigura due personaggi che esplorano il buio dinanzi a loro, aiutandosi l'uno con l'altro, indica un suggerimento utile per chi vuole guidare l'esplorazione di un piccolo principiante (come si vede, il segnale raffigura una tartaruga piccola e una grande). Vi si può trovare la spiegazione di un argomento difficile, o un avvertimento sui trabocchetti più comuni, o anche magari un suggerimento pratico, su come si può aiutare un altro che sta imparando il Logo.
Naturalmente, i suggerimenti valgono anche per chi sta imparando, che trova così una guida e un'occasione per riflettere.

COSA OCCORRE AL LETTORE

Per prima cosa, serve un computer Commodore 64 e un disco del linguaggio Logo. Occorre anche un disco Logo di servizio, per registrare le procedure che man mano si realizzano, lavorando direttamente con il computer.

Sarebbe bene poi tenere un diario, su cui scrivere ciò che si fa giorno per giorno con il Logo, le idee, i piani dei nuovi progetti da costruire, i problemi ancora non risolti e i dubbi che si possono avere su qualche argomento. Conviene anche ricopiare sul diario tutte le procedure fatte, tenerne un elenco aggiornato e descriverne il funzionamento. Se si ha una stampante collegata al computer, la cosa è molto semplice, perché basta stampare le procedure e inserire i fogli nel diario.
È interessante scrivere sul diario anche i bug (o errori) che si sono scoperti, annotando se si è riusciti ad eliminarli o no. Può sempre darsi che qualcuno, ad un certo punto, ci aiuti ad eliminare un vecchio errore. E se si scrive che cosa si è riusciti a correggere da soli, si potrà in seguito aiutare qualcun altro a risolvere lo stesso problema. Non è necessario scrivere moltissimo, ma conviene tenere il diario sempre a portata di mano, per quando succede qualcosa di notevole.

Che cosa è il Logo

Il Logo è per prima cosa un linguaggio di programmazione, con cui si traducono i simboli, i numeri e le parole del linguaggio umano in impulsi elettronici interpretabili dal computer.
Si può anche pensare al Logo come ad un servitore un poco ottuso, ma pieno di risorse, addetto a dirigere il lavoro del computer. Quando si indica un comando da eseguire, il Logo lo traduce per il computer, con la collaborazione dei suoi aiutanti (procedure o comandi). Se un comando risulta poco comprensibile, il Logo lo segnala. È possibile insegnare al Logo come eseguire nuovi comandi, ricorrendo alle cose che già conosce: queste nuove istruzioni sono le cosiddette procedure, ciascuna delle quali mette a disposizione del Logo un nuovo aiutante, specializzato nell'esecuzione di un compito ben preciso.
Viene spontaneo pensare al Logo come se fosse una persona, anche perché rappresenta in effetti un bell'aiuto nel far funzionare il computer, specie in casi complicati o, comunque, impegnativi.
Il Logo, in un senso più ampio, è infine qualcosa di più di uno strumento che si impara ad usare, perché con esso è possibile anche imparare cose nuove. In effetti, usando il Logo per fare eseguire al computer determinati compiti, si approfondisce la conoscenza di ciò che il computer esegue in tal modo, e anche di come noi stessi impariamo cose nuove. Questo libro presenta il Logo come uno strumento di conoscenza.
Per saperne di più sugli aspetti tecnici del Logo, come linguaggio di programmazione, si deve fare riferimento al manuale contenuto nella confezione del Logo. Per approfondire invece il ruolo del Logo nell'apprendimento, i suoi presupposti metodologici e l'uso nell'ambiente educativo,

occorre leggere il libro di Seymour Papert, *Mindstorms: Children, Computers and Powerful Ideas*, che ne spiega lo sviluppo, come strumento capace di rendere il computer un "oggetto con cui pensare". È in ogni caso interessante leggere *Genitori nell'era del computer* di P. Scharf pubblicato dalla McGraw-Hill, che è una rassegna delle problematiche pedagogiche relative all'uso del computer.
Ora si può incominciare!

1

Si incomincia

Qui si impara a caricare il Logo nel computer dal disco del linguaggio Logo, e a usare la tastiera del Commodore. Si fa inoltre conoscenza con un personaggio caratteristico, la tartaruga del Logo, e si vede come farla muovere sullo schermo e come tracciare figure, con alcuni semplici comandi.
Per incominciare sul serio, bisogna avere un computer Commodore 64 e un disco Logo. È meglio, inoltre, tenere a portata di mano il diario del Logo, con una matita o una penna per scriverci le note e i commenti di ciò che si fa. Questi sono i comandi che presenteremo nel capitolo:

Comando	Abbreviazione	Esempi
STAMPA	ST	STAMPA [CATHY PERINI]
DISEGNA		
AVANTI	A	AVANTI 20, A 20
INDIETRO	I	INDIETRO 10, I 10
DESTRA	D	DESTRA 90, D 90
SINISTRA	S	SINISTRA 30, S 30

Non si usano procedure di servizio e neanche file su disco.

1.1 Come caricare il Logo dal disco Logo

Il linguaggio Logo è registrato su disco e non rimane nella memoria del computer quando lo si spegne. Per incominciare, allora, lo si deve carica-

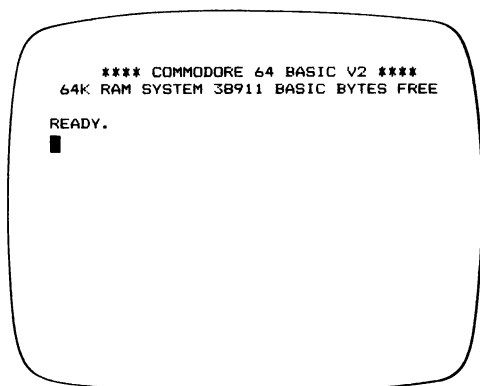



Figura 1.1 Messaggio iniziale del Commodore 64

re in memoria seguendo i seguenti passi:

- Si parte col computer spento, senza dischi nel drive.
- Si accende il drive (quando il drive è acceso, si vede una piccola luce verde).
- Si accende il monitor oppure lo schermo TV.
- Si accende infine il computer (sempre per ultimo).
- A questo punto, il drive si mette a ronzare e si vede sullo schermo il messaggio di Figura 1.1.

TRABOCCHETTO

Questo segnale compare per aiutarvi ad evitare un possibile problema: qui ad esempio può succedere che non si accenda la lucetta verde del drive, oppure che non si veda alcun messaggio visualizzato sullo schermo. Così non si può proseguire: bisogna spegnere tutto e rivedere i collegamenti elettrici. Dopo il controllo, si può riprovare.

Se tutto è in ordine, si procede con questi altri passi:

- Si inserisce il disco Logo nel drive, chiudendo poi lo sportello.
- Si batte (sulla tastiera):

  LOAD "LOGO",8
  concludendo con la battuta del tasto RETURN

- Si dovrebbe allora vedere visualizzato sullo schermo il messaggio:

  SEARCHING FOR LOGO
  LOADING
  READY.

  Si batte quindi:

  RUN (e poi il tasto RETURN)

- Si vede allora sullo schermo la nuova scritta:

  Un momento prego

  Nello stesso tempo si accende la lucetta rossa del drive, e questo si mette a ronzare per un paio di minuti (in questa fase avviene il trasferimento del Logo dal disco alla memoria del Commodore).
- Quando il trasferimento è finito, si vede sullo schermo la scritta di Figura 1.2; si estrae dal drive il disco Logo e lo si ripone al sicuro.

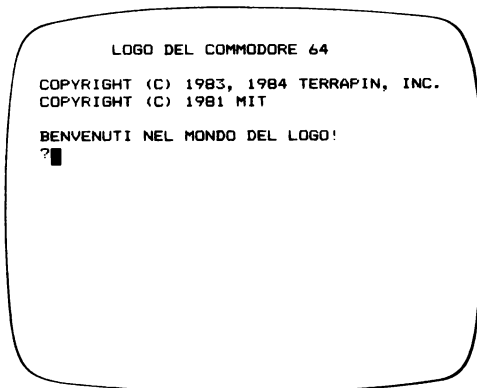



Figura 1.2 Lo schermo del Commodore 64 dopo il caricamento del Logo in memoria

TRABOCCHETTO

Se si incontrano difficoltà nel caricare il Logo, conviene chiedere a qualcuno che ne sa di più, oppure leggere le istruzioni allegate al disco Logo.

COME CI SI FERMA?

Se si vuole smettere, basta mettere da parte il disco Logo, riponendolo con gli altri dischi del Commodore, e poi spegnere tutto.
Se invece qualcun altro è già pronto per provare il Logo, non si spegne il computer, ma si batte:

CIAO (e poi il tasto RETURN)

In questo modo si ripulisce la memoria del computer, senza cancellare il Logo.
Attenzione al tasto RETURN: è un tasto speciale che va battuto quando è indicato (non bisogna battere da tastiera la parola RETURN, ma il tasto segnato sulla tastiera, appunto, con RETURN).

1.2 La tastiera del Commodore 64

A questo punto bisogna considerare la tastiera e il suo funzionamento. In effetti, la tastiera di un computer è molto simile a quella di una macchina da scrivere: quando si batte un tasto, il simbolo corrispondente compare sullo schermo, invece che essere stampato su carta. Il quadratino lampeggiante che si vede sempre sullo schermo si chiama *cursore*, e indica la posizione in cui comparirà la lettera o il numero o il simbolo successivo che verrà battuto alla tastiera.
Osservando la Figura 1.3, che riproduce la tastiera del Commodore 64, si notano alcuni tasti speciali, che vanno conosciuti per poter imparare il Logo.

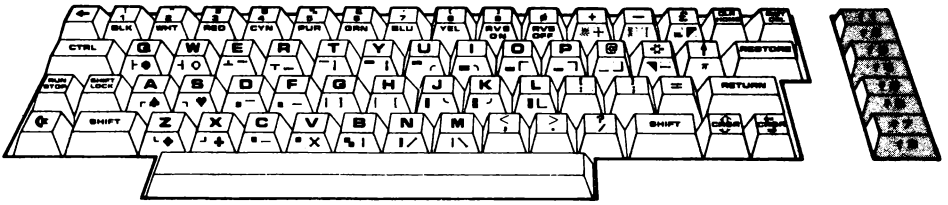
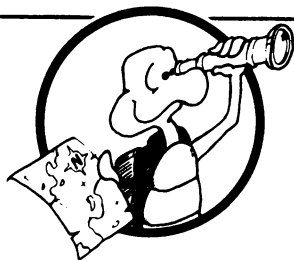



Figura 1.3 La tastiera del Commodore 64, con i tasti alfabetici, numerici e speciali

TASTI SPECIALI

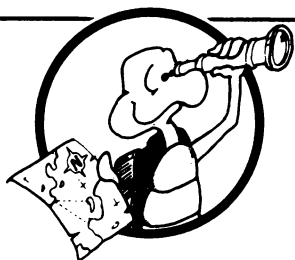
I tasti speciali sono quelli contrassegnati con INST/DEL, CTRL, SHIFT, RETURN, ⇑CRSR⇓ e ⇐ CRSR⇒ .
Il tasto INST/DEL fa fare un passo indietro al cursore cancellando la lettera o il numero appena battuto. Se si tiene premuto questo tasto, la manovra ha luogo più volte, fino a che non si rilascia il tasto. È bene fare attenzione per non cancellare troppo.
Il tasto CTRL serve in combinazione con altri tasti, per specificare comandi speciali. Ad esempio CTRL-G indica la combinazione del tasto CTRL col tasto G, e serve a far fermare il Logo, qualunque cosa stia facendo. In casi come questo, si preme per primo il tasto CTRL, battendo poi l'altro tasto che si combina con esso, nel comando.
Sulla tastiera, il tasto SHIFT compare due volte: anch'esso è usato in combinazione con altri tasti, ad esempio SHIFT-1 produce il simbolo ! e SHIFT-/ produce il simbolo ?.
Inoltre, SHIFT-[e SHIFT-] producono i simboli [e], rispettivamente (questi sono due segni molto importanti nel Logo, come pure il segno ", che si ottiene con la combinazione SHIFT-2).
Il tasto RETURN è fondamentale per il Logo, perché è il segnale per dire al Logo che il comando appena battuto va eseguito. In effetti, ogni comando del Logo si deve concludere con il tasto RETURN.
Il tasto ⇐CRSR⇒ serve per muovere il cursore verso destra o (quando si combina con il tasto SHIFT) verso sinistra. Anche il tasto CRSR funziona a ripetizione, se lo si tiene premuto.
Il tasto contrassegnato da ← serve anch'esso a muovere il cursore, ma solo di un passo verso sinistra, e senza la ripetizione automatica.
I tasti che, se tenuti premuti, funzionano a ripetizione sono i seguenti:

INST/DEL
i due tasti CRSR
la barra spaziatrice

ESPLORAZIONE

Questo segnale compare quando è il momento di fare un po' da soli, lasciando il libro per provare direttamente che cosa fa il computer. Il modo migliore per imparare è provare, e non c'è da aver paura di sbagliare. A questo punto si esplora la tastiera. Ecco alcuni modi:

- Battere molti tasti, e poi provare il tasto INST/DEL, per vedere se davvero cancella i simboli sullo schermo, e come.
- Battere CTRL-G (in pratica, si batte il tasto G mentre si tiene premuto il tasto CTRL).
- Impiegare il tasto SHIFT, combinandolo con molti altri tasti. Provare ad esempio a produrre le parentesi quadre, con SHIFT-[e con SHIFT-], e vedere se davvero si ottiene " battendo SHIFT-2. Questi simboli sono fondamentali per alcuni comandi del Logo.

TRABOCCHETTO

Per usare correttamente la combinazione di tasti CTRL-G oppure SHIFT-[occorre prima premere il tasto speciale (CTRL o SHIFT) e poi, tenendolo premuto, battere il secondo tasto G o [o comunque quello previsto dal comando che si intende dare.
Quando si è prodotto l'effetto voluto, si rilasciano entrambi i tasti.

ESPLORAZIONE

Ecco che cosa si può, ad esempio, esplorare sulla tastiera:

- Si batte il tasto RETURN da solo. Il Logo può scrivere un messaggio di protesta, che può apparire strano, per ora (più avanti si vedrà come interpretare i messaggi di protesta del Logo).
- Si batte una qualunque successione di tasti, uno dopo l'altro. Poi si impiega il tasto CRSR per spostare il cursore. Conviene notare la differenza fra il tasto CRSR e il tasto INST/DEL: entrambi spostano il cursore a sinistra (il tasto CRSR va usato combinato col tasto SHIFT, se no sposta il cursore verso destra), però solo il tasto INST/DEL cancella i caratteri che incontra. Anche la barra di spaziatura e il tasto CRSR (da solo, senza il tasto SHIFT) hanno effetto diverso, pur spostando il cursore verso destra: infatti solo la barra di spaziatura lascia uno spazio vuoto.

1.3 Come battere i comandi del Logo

Quando sullo schermo compare la scritta BENVENUTI NEL MONDO DEL LOGO!, si è pronti per incominciare. Il simbolo ? che compare sotto la parola BENVENUTI è il modo con cui il Logo segnala che è pronto ad eseguire i comandi. In pratica, è come se il Logo ci dicesse: sono qui, cosa vuoi da me?
Il quadratino lampeggiante a fianco del simbolo ? è il cursore, che serve da guida per la battitura dei comandi. Esso indica il punto dove comparirà il successivo carattere battuto alla tastiera (un carattere può essere una lettera, un numero o un segno speciale, di quelli presenti sulla tastiera). Il cursore è molto utile anche quando si vuole modificare qualcosa che si è già battuto, come si vedrà più in là.
Si può provare ora a battere una scritta semplice, ad esempio un nome. Proviamo allora a battere Cathy Perini (oppure anche il proprio nome, o qualunque altra frase). Si batte allora CATHY PERINI e, poi, il tasto RETURN (che, come si è già visto, significa per il Logo "avanti, fai quello che è scritto").
Dopo la battuta del tasto RETURN, sullo schermo compare la seguente scritta:

?CATHY PERINI
LA PROCEDURA CATHY NON ESISTE
?

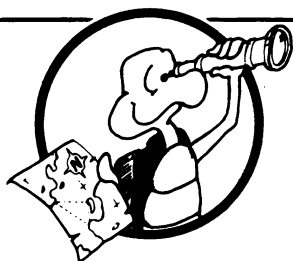
Quando il Logo scrive sullo schermo il segno ? si aspetta un comando, qualcosa insomma da fare.
Se allora si batte, ad esempio:

STAMPA [CATHY PERINI] (e RETURN)

si dovrebbe ottenere sullo schermo questa scritta:

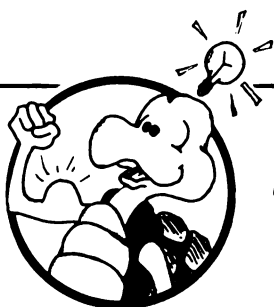
?CATHY PERINI
LA PROCEDURA CATHY NON ESISTE
?STAMPA [CATHY PERINI]
CATHY PERINI
?

TRABOCCHETTO

Ci si può dimenticare di usare la combinazione corretta dei tasti SHIFT-[e SHIFT-] per produrre le parentesi quadre.
Bisogna sempre tener presente che RETURN significa il tasto segnato RETURN sulla tastiera e non la parola RETURN.
Nell'ultimo esempio, STAMPA segnala al Logo di scrivere sullo schermo ciò che segue la parola STAMPA stessa, racchiuso fra parentesi quadre. La scritta CATHY PERINI, nell'esempio proposto, è l'argomento del comando STAMPA. Il Logo è anche molto pignolo sui comandi, e se trova qualche imprecisione protesta senz'altro.
Se si commette un errore di battitura, si può sempre ricorrere al tasto INST/DEL che cancella i caratteri sbagliati, al posto dei quali si possono poi battere i caratteri corretti.

ESPLORAZIONE

Proviamo gli esempi seguenti, per vedere quali fanno protestare il Logo:

STAMPA (e RETURN)
STAMPA CATHY PERINI (e RETURN)
STAMPACATHYPERINI (e RETURN)
STAMPA[CATHY PERINI] (e RETURN)

Naturalmente, il nome CATHY PERINI è solo un esempio, che può essere sostituito con un qualunque altro nome. E bene esplorare a fondo il comando STAMPA, cercando di scrivere sullo schermo le frasi più disparate. Conviene anche far protestare il Logo, apposta per vedere che cosa scrive in questi casi (sono i cosiddetti messaggi di errore).

IDEA

Questo segnale compare per indicare idee interessanti e soprattutto utili, che rendono più capace e potente chi le comprende.
Qui, ad esempio, conviene capire che non si può danneggiare in alcun modo il computer, con comandi sbagliati: si può battere quel che si vuole. Certo, il Logo è molto pignolo sui comandi, e per ottenere qualcosa bisogna fare attenzione a battere il comando nel modo giusto, altrimenti non capisce e protesta.

SUGGERIMENTO

Sotto questo segno si trovano consigli utili a chi intende aiutare un principiante, o vuole avere un'occasione in più per riflettere durante l'apprendimento del Logo.
È utile, quando si incomincia ad imparare il Logo, farlo protestare. Si riesce così a scoprire che cosa non accetta e a vedere molti messaggi di errore. Inoltre un principiante può in tal modo rendersi conto che il Logo va trattato a modo suo.
Anche chi ormai conosce bene il Logo e lo usa spesso può commettere errori di battitura, o imprecisioni di sintassi, perché magari non rispetta esattamente le regole fissate per esprimere i comandi. Il Logo risponde a comandi ben determinati e solo a quelli, ed ha una sintassi non arbitraria, ma rigida. In effetti, ci sono ragioni ben precise per cui il comando STAMPA [CATHY PERINI] funziona bene e invece il comando STAMPA CATHY PERINI non va. Più avanti si potranno capire, e per ora conviene assumere che i comandi siano come parole d'ordine stabilite da altri, a loro piacere, da rispettare senza far domande. Il modo migliore per imparare i comandi è provarli direttamente, magari sbagliando (così intanto ci si abitua all'idea che esistono comandi scritti bene e comandi scritti male).

1.4 La tartaruga

Dopo aver imparato come si danno i comandi al Logo, si è pronti per l'incontro con la tartaruga, cioè con l'elemento usato nella maggior parte delle attività grafiche. Conviene senz'altro battere il comando:

DISEGNA (e RETURN)

Il comando DISEGNA dice in pratica al Logo: "Ripulisci lo schermo, e fai comparire la tartaruga al centro, con la testa rivolta in alto".
Si può muovere la tartaruga con i comandi AVANTI e INDIETRO. Inoltre, la si può girare con i comandi DESTRA e SINISTRA. Prima di muovere o di girare la tartaruga, bisogna sempre dire di quanto la si intende muovere (o girare), indicandolo con un numero.
Si può provare ora con i comandi riportati qui sotto, facendo attenzione a lasciare uno spazio fra il comando e il numero che lo riguarda, altri-

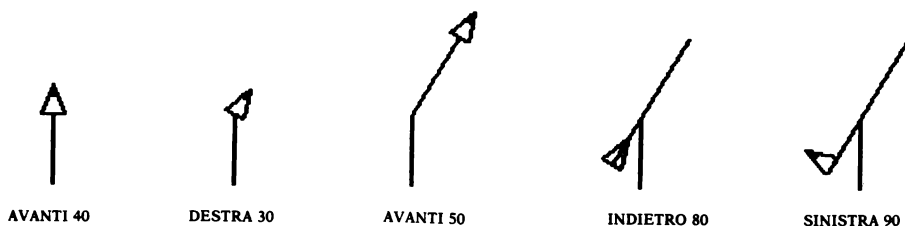



Figura 1.4 Effetto di acuni comandi di manovra della tartaruga

menti il Logo protesta. Bisogna anche ricordarsi di battere alla fine di ogni comando il tasto RETURN, altrimenti il Logo non si muove e continua ad aspettare, come se niente fosse.
Ecco i comandi da provare:

```
AVANTI 40
DESTRA 30
AVANTI 50
INDIETRO 80
SINISTRA 90
```

Ora si può battere nuovamente il comando DISEGNA, per ripulire lo schermo.

L'immagine della tartaruga in questo libro è leggermente differente da quella che compare sullo schermo del Commodore. Qui il naso della tartaruga è scuro, e indica la direzione verso cui essa è rivolta, mentre la posizione di questa è data dal centro del lato posteriore.
Sul Commodore invece è scuro il lato posteriore, e la posizione della tartaruga è quella del suo centro.

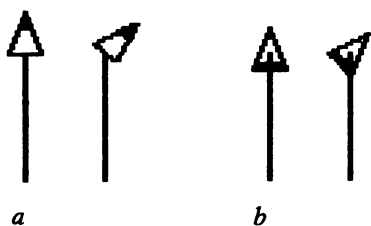
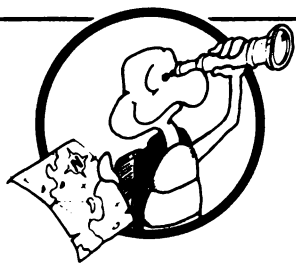



Figura 1.5 La tartaruga come appare nel libro (*a*), e come appare sullo schermo del Commodore (*b*)

Per fare eseguire un comando al Logo, bisogna sempre concluderlo con il tasto RETURN. Se poi si batte un comando, come AVANTI, senza specificare il numero, oppure si scrive il comando e il numero di seguito, come in DESTRA30, non si ottiene nulla e il Logo protesta. (Anzi, perché non provare a farlo protestare?)

I comandi possono essere dati anche in forma abbreviata, come si vede qui di seguito:

A 40
D 30
A 50
I 80
S 90

ESPLORAZIONE

Nel prossimo capitolo si troveranno moltissime occasioni per sperimentare la tartaruga e le sue manovre. Qui conviene provare qualche semplice manovra, per capire bene i comandi:

- Muovere la tartaruga sullo schermo, con i comandi AVANTI, INDIETRO, DESTRA e SINISTRA, assegnando numeri ogni volta diversi.
- Ripulire lo schermo con il comando DISEGNA, e far tracciare alla tartaruga un percorso bizzarro, scelto a caso.
- Fissare un punto sullo schermo e muovere la tartaruga fin là.
- Quanto è distante il centro dello schermo dal bordo superiore e dai lati? Usare la tartaruga per scoprire queste distanze.
- Battere i comandi AVANTI, INDIETRO, DESTRA e SINISTRA con numeri molti grandi. Riprovare, questa volta con numeri molto piccoli. Quali sono le distanze più grandi e più piccole che può percorrere la tartaruga?

SUGGERIMENTO

Lo scopo di questo primo capitolo è di prendere confidenza con il computer, con la tastiera e con la forma dei comandi del Logo. Un buon aiuto può venire da un elenco dei comandi di manovra della tartaruga, che si può disporre vicino al computer. Man mano che si apprendono nuovi comandi, si può ampliare l'elenco. Il vantaggio è evidente, perché così chi impara può fare a meno di sfogliare in continuazione il libro, per vedere come si scrive esattamente un certo comando o per controllare se occorre assegnare un numero.

Se ci si trova in una scuola, o in un club dove il computer viene usato da molti, conviene mettere bene in vista le istruzioni per caricare in memoria il Logo dal disco, e per eseguire altre operazioni pratiche, come l'accensione e lo spegnimento del computer, la tenuta ordinata dei dischi, e così via. Se vicino al computer si trovano istruzioni operative chiare, i ragazzi si possono aiutare l'uno con l'altro, risparmiando molta fatica all'insegnante.

2

Il mondo della tartaruga

In questo capitolo si esplora la tartaruga, come se fosse un nuovo pianeta in cui orientarsi, apprendendo nuovi comandi con cui dirigerla e con cui realizzare anche disegni a colori.
Una volta fatto un disegno che piace, conviene per il momento trascriverne i comandi sul diario del Logo. Più avanti, nel Capitolo 4, vedremo il modo di assegnare al computer nuovi comandi e il modo di registrarne le istruzioni sul disco di servizio, dedicato al Logo.

Comando	Abbreviazione	Esempi
ESCIDIS	ED	
SULAPENNA	SU	
GIULAPENNA	GIU	
ASCOL		ASCOL 3
COLORESFONDO	CS	COLORESFONDO 56, CS 5
RECUPERA		RECUPERA "CERCHI
NASTARTA	NT	
MOSTARTA	MT	
ATTORNO		
MURO		
PULISCISCHERMO	PS	
TANA		
TESCHERMO	f1	
DISCHERMO	f3	
GRASCHERMO	f5	

Viene usata la procedura "CERCHI dell'Appendice A.

Procedure del disco di servizio utilizzate:

Procedura	Esempi
CERCHIOD	CERCHIOD 20
CERCHIOS	CERCHIOS 10
ARCOD	ARCOD 40
ARCOS	ARCOS 20

2.1 I comandi elementari della tartaruga

Per imparare cosa può fare la tartaruga, conviene senz'altro provarla in pratica. Ecco qui alcuni comandi con cui incominciare l'esplorazione del nuovo mondo della tartaruga:

DISEGNA
Colloca la tartaruga al centro dello schermo, dopo averlo ripulito. Il comando DISEGNA è il più comodo per ricominciare da capo, con uno schermo pulito.

ESCIDIS (o anche ED)
Fa sparire la tartaruga, lasciando libero l'intero schermo per la visualizzazione di ciò che si batte da tastiera.

AVANTI e INDIETRO (abbreviati in A e I)
Spostano la tartaruga in avanti e all'indietro: bisogna però indicare la distanza dello spostamento (battendo un numero), come si vede in questi esempi:

AVANTI 20 (oppure A 20)
INDIETRO 50 (oppure I 50)

DESTRA e SINISTRA (abbreviati anche in D e S)
Ruotano la tartaruga verso sinistra, o verso destra. Anche qui va specificato un numero, che indica l'angolo di rotazione, come in questi esempi:

DESTRA 45 (oppure D 45)
SINISTRA 100 (oppure S 100)

TRABOCCHETTO

Attenzione, a lasciare sempre uno spazio vuoto fra il nome del comando e il numero che specifica la distanza, o l'angolo. Se si scrive ad esempio AVANTI100 oppure D30, il Logo non capisce e protesta.

Ecco ancora altri comandi:

SULAPENNA (ovvero SU)
Solleva il pennino della tartaruga, in modo che questa non lasci traccia del suo passaggio. Il pennino rimane sollevato fino a che non lo si abbassa, con il comando GIULAPENNA.

GIULAPENNA (ovvero GIU)
Abbassa il pennino della tartaruga, che così lascia traccia del suo passaggio.

Questi due comandi servono, ad esempio, quando si vuole far iniziare un disegno da un punto che non sia il centro dello schermo, o anche quando si tracciano figure separate, o una interna all'altra.

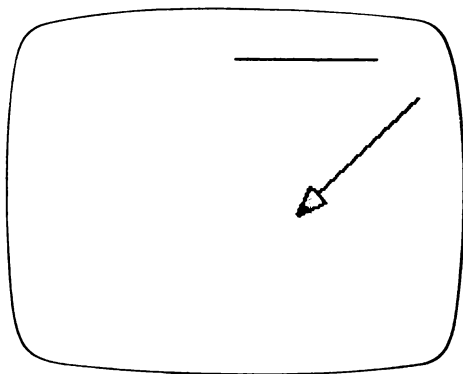



Figura 2.1 Impiego dei comandi SULAPENNA e GIULAPENNA

Conviene fare un po' di pratica, con i comandi appena visti, incominciando con lo schermo pulito:

```
DISEGNA
SULAPENNA
AVANTI 50
DESTRA 90
GIULAPENNA
AVANTI 50
SULAPENNA
DESTRA 45
AVANTI 20
GIULAPENNA
DESTRA 90
AVANTI 50
```

Con i comandi SULAPENNA e GIULAPENNA si ottengono effetti come quelli illustrati in Figura 2.2.

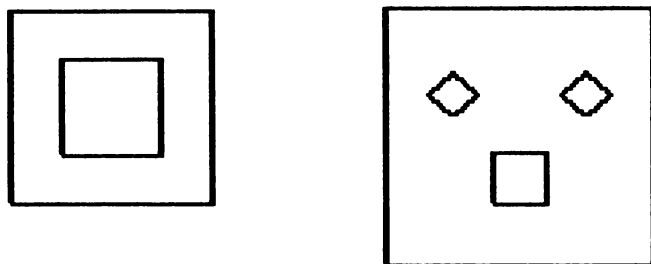
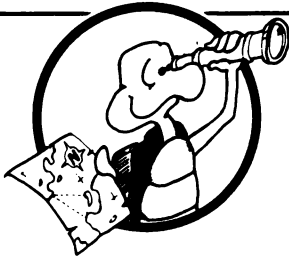



Figura 2.2 Disegni costruiti con l'impiego dei comandi SULAPENNA e GIULAPENNA

2.2 Alla scoperta della tartaruga

La tartaruga va conosciuta bene, perché accompagna sempre chi si serve del Logo: conviene dedicare sin dall'inizio tutto il tempo necessario alla sua conoscenza, fino a prendere confidenza.

ESPLORAZIONE

Ecco alcune idee per incominciare l'esplorazione della tartaruga:

- Scegliere un punto sullo schermo, spostando lì la tartaruga. Riprovare con diversi punti.

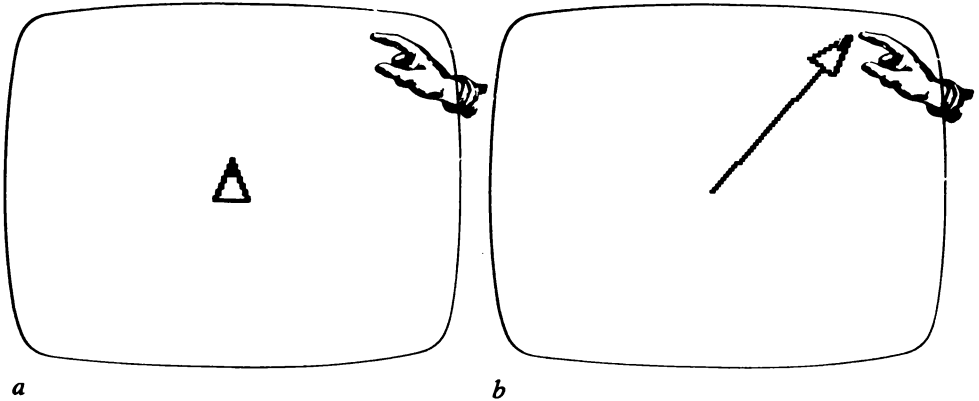



Figura 2.3 Scegliere un punto sullo schermo (*a*) e muovere la tartaruga verso di esso (*b*)

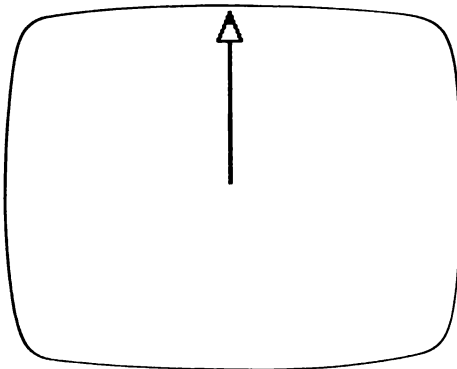



Figura 2.4 Come trovare la distanza fra il centro e il bordo dello schermo

- Qual è la distanza fra il centro e il bordo dello schermo? Usare la tartaruga per trovare la risposta. Cosa succede se la tartaruga oltrepassa il bordo? Si riesce a farla ritornare?
- Battere il comando DISEGNA per ripulire lo schermo. Voltare la tartaruga verso destra: di quanto bisogna ruotarla perché si trovi in direzione orizzontale? Per vedere se è proprio orizzontale, la si sposta in avanti, e si osserva la traccia.

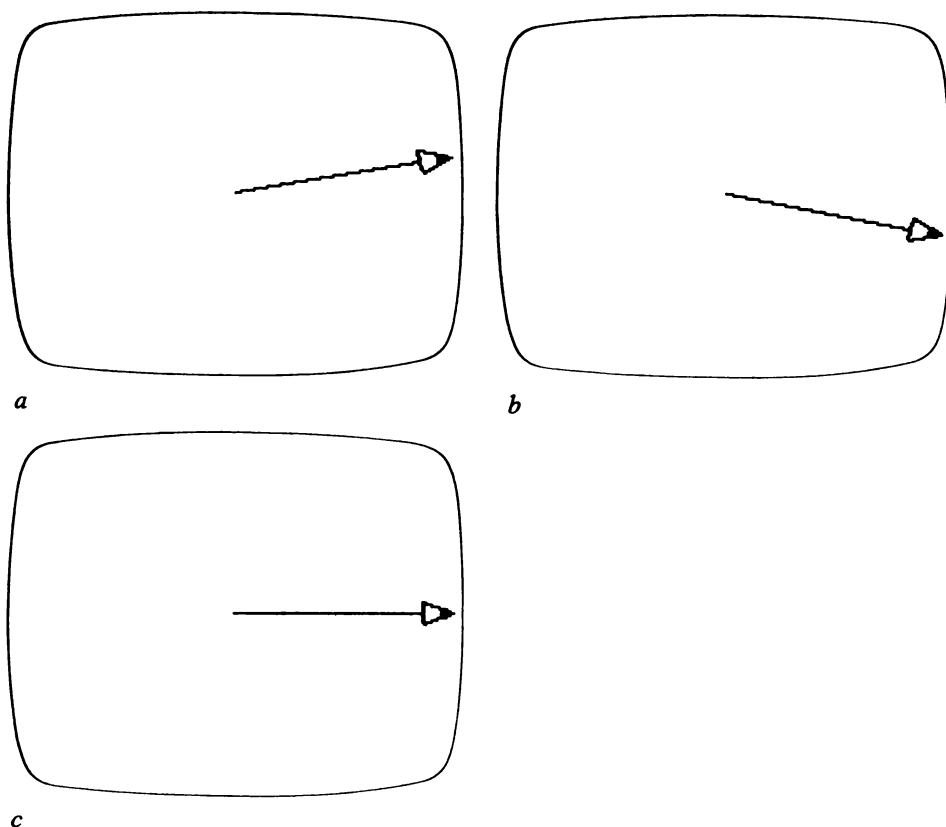



Figura 2.5 Esercizi di rotazione della tartaruga: (*a*) ruotata troppo poco; (*b*) ruotata troppo; (*c*) ruotata esattamente in posizione orizzontale

- Trovare, con la tartaruga, quanto è largo lo schermo.
- Fare oltrepassare alla tartaruga il bordo dello schermo. Si riesce a fare un giro completo e a farla ritornare al punto di partenza, ma dalla direzione opposta? Quanto deve essere lo spostamento?

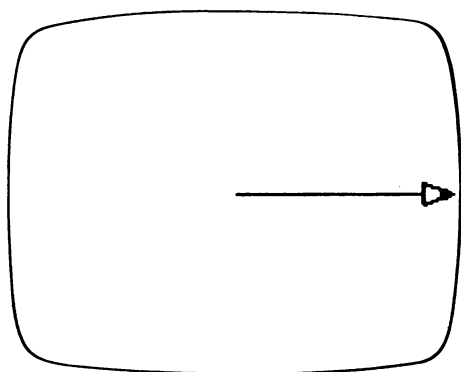



Figura 2.6 Ecco come si prova la distanza fra il centro e il bordo dello schermo

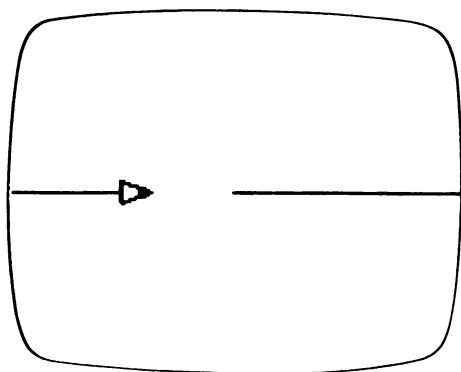



Figura 2.7 Ecco la tartaruga che ha fatto il giro attorno allo schermo

Quando la tartaruga fa il giro completo dello schermo, scomparendo oltre un bordo e ricomparendo dall'altro, si dice anche che si avvolge intorno allo schermo (più o meno come fa un nastro intorno a un pacco).

- Battere il comando DISEGNA, e ruotare appena appena la tartaruga. Battere poi il comando AVANTI, specificando uno spostamento a grande distanza. Se lo spostamento è grande abbastanza, si vede l'effetto di Figura 2.8.

a *b*

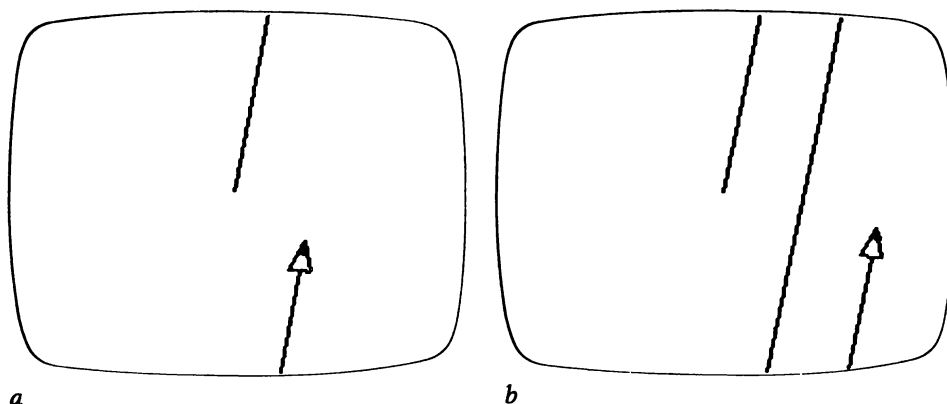



Figura 2.8 Linea tracciata dalla tartaruga, con un avvolgimento (*a*), e con più avvolgimenti (*b*) intorno allo schermo

TRABOCCHETTO

Qualche volta, la tartaruga sembra scomparire dietro le righe della parte bassa dello schermo, in cui compaiono varie scritte: niente paura, anche se non si vede, la tartaruga è proprio lì.
Se si vuole vedere tutto quanto lo schermo, senza le scritte di fondo, basta battere il comando F5.
Un altro comando speciale, F3, ripristina la suddivisione dello schermo, quando ad esempio si vuole battere un nuovo comando.

SUGGERIMENTO

Considerazioni sul comportamento esplorativo e sulla scoperta

L'esplorazione è un momento importante di conoscenza del modo di funzionare della tartaruga. Molti amano entrare in nuovi argomenti, provando tutte le idee che vengono loro in mente, mentre altri preferiscono essere comunque guidati, per avere una traccia sicura da seguire (ad esempio, nel caso della tartaruga, uno o più disegni già stabiliti, da riprodurre). L'apprendimento del Logo si basa, in effetti, su entrambi i modi di apprendimento, l'esplorazione spontanea e la costruzione guidata, che vanno alternati e integrati in modo opportuno. Sovente, anzi, accade che un'esplorazione conduca a trovate interessanti, che tornano utili nelle costruzioni finalizzate da realizzare in seguito.

Per esempio, consideriamo il disegno di un quadrato. Ciò che bisogna conoscere è l'angolo di cui ruotare la tartaruga prima di tracciare un nuovo lato. Ora, questo angolo è anche quello di cui deve ruotare la tartaruga per tracciare una linea orizzontale attraverso lo schermo, e può essere scoperto durante un'esplorazione libera, rivelandosi per quello che è, cioè l'angolo di 90 gradi, e per le sue relazioni con altri angoli caratteristici (un quarto dell'angolo di 360 gradi, che fa compiere alla tartaruga un giro completo su se stessa, il doppio dell'angolo di 45 gradi, che traccia la diagonale, ecc.). L'esplorazione si rivela molto utile anche perché abitua chi impara a considerare l'aspetto numerico del fenomeno, e a notare proprietà e fatti caratteristici.

Fra l'altro, con l'attività di esplorazione si può a questo punto raggiungere una buona familiarità con la tastiera e con i movimenti della tartaruga (in avani, all'indietro, a destra, a sinistra), e si riesce a stimare correttamente distanze e angoli di rotazione sullo schermo. Si ottiene lo stesso risultato, in modo più sistematico, con il gioco TIRO.A.SEGNO presentato nel Capitolo 3.

Più importante ancora delle risposte alle varie domande è il metodo di indagine che si impara con l'attività esplorativa. Ad esempio, la vera lezione da ricordare non è tanto "la distanza tra centro e bordo dello schermo", quanto il fatto che è possibile rispondere alla domanda servendosi del computer. Le domande poste ammettono in genere più d'una risposta, e più d'un metodo di soluzione. Se si ha a che fare con un gruppo di allievi, è bene chiedere a ciascuno di presentare la propria soluzione, e incoraggiare poi tutti a discutere insieme le risposte e i metodi impiegati per trovarle. Se si riesce a inculcare l'idea di discutere in gruppo le risposte e il ragionamento fatto per trovarle (accettando di esporsi magari alla contestazione degli altri), si può considerare raggiunto lo scopo dell'attività esplorativa qui proposta.

Se si ha un gruppo di allievi, conviene anche predisporre un bel quaderno,

in cui gli allievi possano scrivere le idee, le domande, le scoperte e i suggerimenti di ciascuno. Questo può aiutare a creare un ambiente collaborativo, in cui gli allievi discutono liberamente e si aiutano a vicenda.

2.3 Come tracciare figure con la tartaruga

Con la tartaruga si possono disegnare moltissime figure. Combinando le più semplici (ad esempio, a forma di quadrato) si possono inoltre ottenere disegni complessi, come si vede in Figura 2.9.

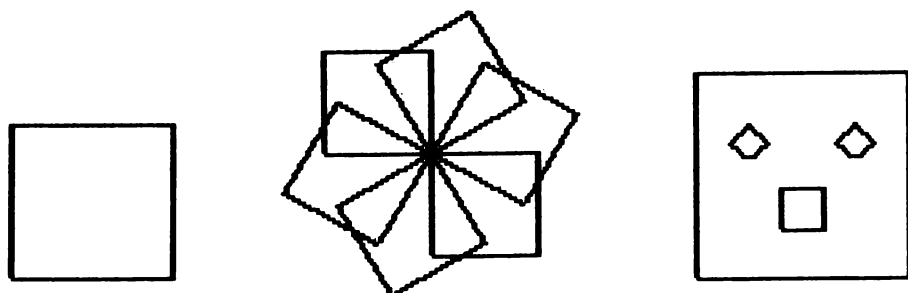
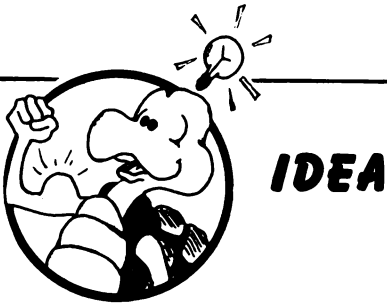



Figura 2.9 Disegni costruiti con quadrati, tracciati con la tartaruga

Si vedrà presto come fare per combinare più disegni: per ora si può provare il caso di figure semplici, a forma di quadrato o di rettangolo (il quadrato è semplicissimo, avendo gli angoli e i lati uguali fra loro).
Un buon metodo per imparare in fretta a guidare la tartaruga è immaginare di essere nei suoi panni e di camminare lungo la figura che si intende tracciare, pensandola già tracciata sul pavimento. Allora, per un quadrato, si immagina di andar dritto per cinque passi, poi di svoltare a destra di 90 gradi, e di ripetere il tutto finché non si giunge ancora al punto di partenza.

Sì, per risolvere molti problemi di costruzione grafica conviene immedesimarsi nella tartaruga, e cercare di riprodurre mentalmente i movimenti specificati nei vari comandi di manovra che ci vengono in mente. E poi, così ci abituiamo a trattare Turtle da amico.

Ora si può provare a disegnare un quadrato: non è difficile. I primi passi sono questi (dopo aver ripulito lo schermo con il comando DISEGNA):

```
AVANTI 50
DESTRA 90
AVANTI 50
```

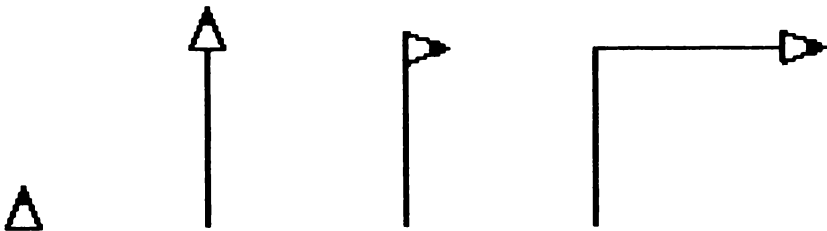
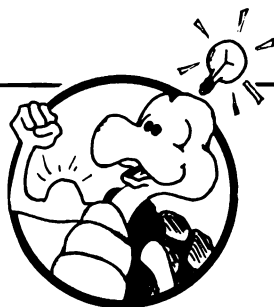



Figura 2.10 Primi passi di costruzione di un quadrato

E ora, coraggio, si completa il quadrato.

C'è un modo molto comodo di mettersi nei panni di Turtle, senza doversi alzare dalla sedia e camminare su e giù per la stanza. Basta ruotare la testa (e i piedi) in modo da allinearli con la tartaruga sullo schermo: così è più facile decidere quali movimenti vanno fatti, e a quale distanza.

Ecco ora altri esempi interessanti:

- Costruire un quadrato grande e uno piccolo.
- Costruire un quadrato girando a sinistra.
- Ruotare la tartaruga e costruire poi un quadrato inclinato.

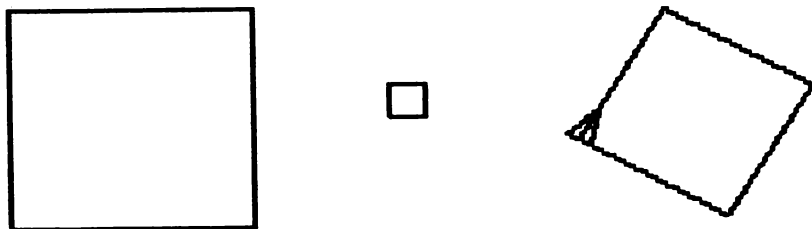



Figura 2.11 Quadrati

- Provare con i rettangoli (che sono diversi dai quadrati solo perché i lati non sono tutti e quattro uguali, ma lo sono solo a due a due: i quattro angoli, invece, sono sempre uguali). Se sembra difficile tracciare un rettangolo, ci si può alzare in piedi e si prova a percorrerne uno, nella stanza.
- Provare a disegnare le proprie iniziali.
- A questo punto si possono disegnare figure di fantasia (in Figura 2.14 ci sono alcuni esempi).

Se si vuole insegnare al computer un nuovo comando, per aiutarlo a tracciare una particolare figura, bisogna passare al Capitolo 4. Poi però conviene ritornare qui e proseguire con il resto del Capitolo 2.

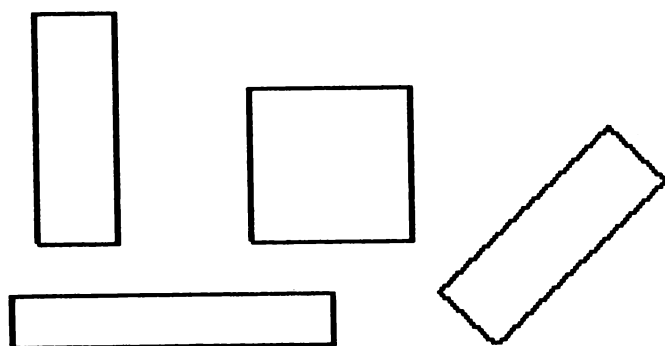



Figura 2.12 Rettangoli

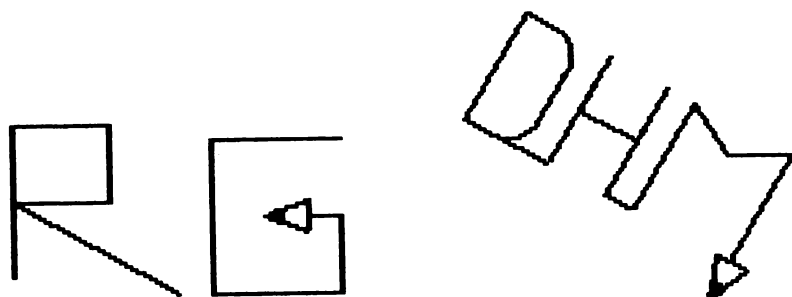



Figura 2.13 Iniziali tracciate con la tartaruga

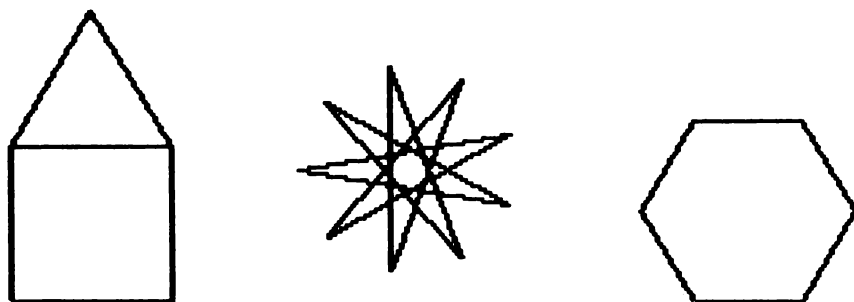



Figura 2.14 Figure di fantasia tracciate con la tartaruga

SUGGERIMENTO

Nei panni di Turtle

L'identificazione con Turtle è molto importante, perché così chi impara può ricorrere a una delle conoscenze più radicate e istintive che possiede, quella del movimento del proprio corpo. Con questo tipo di riferimento, è possibile rendere concreto qualunque problema di manovra della tartaruga, che viene quindi risolto in modo pratico e senza particolari difficoltà.
"Cosa farei se fossi Turtle?" è una domanda che risolve già metà del problema, molto più che la domanda "Come posso dirigere la tartaruga sul piano dello schermo?". Al limite, è possibile riprodurre materialmente i movimenti della tartaruga, camminando su e giù per la stanza e poi trasferire la soluzione sullo schermo. Vale la pena di provare.
Invece di riprodurre materialmente i movimenti della tartaruga, ci si può limitare al pensiero, ruotando anche il capo (e magari pure le spalle) nella direzione verso cui punta la tartaruga, sullo schermo. Questa simulazione mentale del movimento della tartaruga serve soprattutto per decidere quale dovrebbe essere il passo successivo: presto si scopre che non è quasi mai necessario mettersi davvero a camminare, per mettersi nei panni di Turtle.
Per un gruppo di ragazzi, si può anche organizzare un gioco insieme utile e piacevole, in cui uno fa la parte di Turtle, e va su e giù per la stanza, o in cortile, mentre gli altri lo dirigono con i comandi tipici di Logo.
In questo modo si riesce spesso a cementare lo spirito di gruppo e di collaborazione, oltre che a toccare con mano il modo di funzionare della tartaruga. Bisogna però stabilire regole di manovra molto chiare per tutti, in modo che comandi come AVANTI 10 o DESTRA 30 vengano intesi da tutti nello stesso modo. Se si esprimono le rotazioni con angoli, conviene limitare la scelta ai valori più semplici, come ad esempio ai multipli di 90, o di 45, o di 30 gradi, in modo che la persona che interpreta Turtle possa girarsi in una fra le quattro (oppure otto, oppure dodici) direzioni diverse. In effetti, qui non è importante la precisione delle distanze e degli angoli, ma la chiarezza delle scelte per tutti i partecipanti.
Per una buona riuscita didattica del gioco, è importante stabilire che Turtle può solo eseguire i comandi che le vengono dati, senza poterli interpretare in nessun modo. Se ad esempio il comando è di camminare avanti, così deve essere fatto, senza deviazioni verso destra o sinistra, anche se si capisce che la direzione indicata non è quella giusta.
Se si gioca per la prima volta con un nuovo gruppo, conviene che sia l'insegnante a fare la parte di Turtle. Ciò comporta diversi vantaggi. Intanto, i ragazzi si divertono a dirigere i movimenti dell'insegnante, che può dal canto suo dare l'esempio del modo in cui va interpretato Turtle (che è un servitore obbediente in tutto, e completamente privo di spirito critico). Inoltre que-

sto gioco può sembrare, la prima volta, un poco sciocco, e se l'insegnante impersona per primo Turtle, gli allievi perdono l'imbarazzo quando tocca il loro turno.
Per quanto riguarda il secondo modo di mettersi nei panni di Turtle, quello simulato, si può osservare che risulta di grande aiuto soprattutto per chi non ha ancora familiarità nel distinguere la destra dalla sinistra. Esso è inoltre un metodo molto diffuso in pratica, quando si tratta di manovrare la tartaruga in casi particolarmente complicati. Qualche volta, anzi, chi simula in questo modo il movimento di Turtle sente il bisogno di alzarsi e di riprodurre materialmente il movimento della tartaruga.

Dove guarda Turtle?

Per qualcuno risulta a volte difficile capire in quale direzione è puntata la tartaruga. Esiste, in effetti, una differenza fra il modo in cui il computer calcola la direzione (cioè l'angolo di rotazione della tartaruga) e il modo in cui questa viene rappresentata sullo schermo TV, con la tecnica della cosiddetta grafica ad alta risoluzione. L'immagine sullo schermo non è definita così finemente da poter rappresentare esattamente gli angoli. Comunque, è sempre utile tener presente che la tartaruga punta nella direzione dell'angolo tratteggiato in grigio, del triangolino che la rappresenta. Per lo stesso motivo accade che rotazioni molto piccole (come ad esempio DESTRA 1, oppure anche DESTRA 3) non hanno alcun effetto visibile sulla posizione della tartaruga all'interno dello schermo: su di esso infatti si possono rappresentare solo 36 posizioni diverse della tartaruga. Comunque, le linee tracciate rispettano l'orientamento indicato. Conviene provare i casi qui di seguito indicati, per vedere che cosa succede in pratica:

```
DISEGNA
AVANTI 50
INDIETRO 50
DESTRA 4
```

Dopo il comando DESTRA 4, pare che la tartaruga non si sposti affatto.

```
INDIETRO 50
```

La nuova linea, comunque, è tracciata nella direzione giusta (vedi Figura 2.15).

2.4 Il colore

Se si ha a disposizione un apparecchio TV o un monitor a colori, si può disegnare con la tartaruga usando 16 colori diversi, facendo ricorso al comando ASCOL per scegliere di volta in volta il colore desiderato. Il colore che effettivamente si ottiene dipende anche da come è stato imposta-

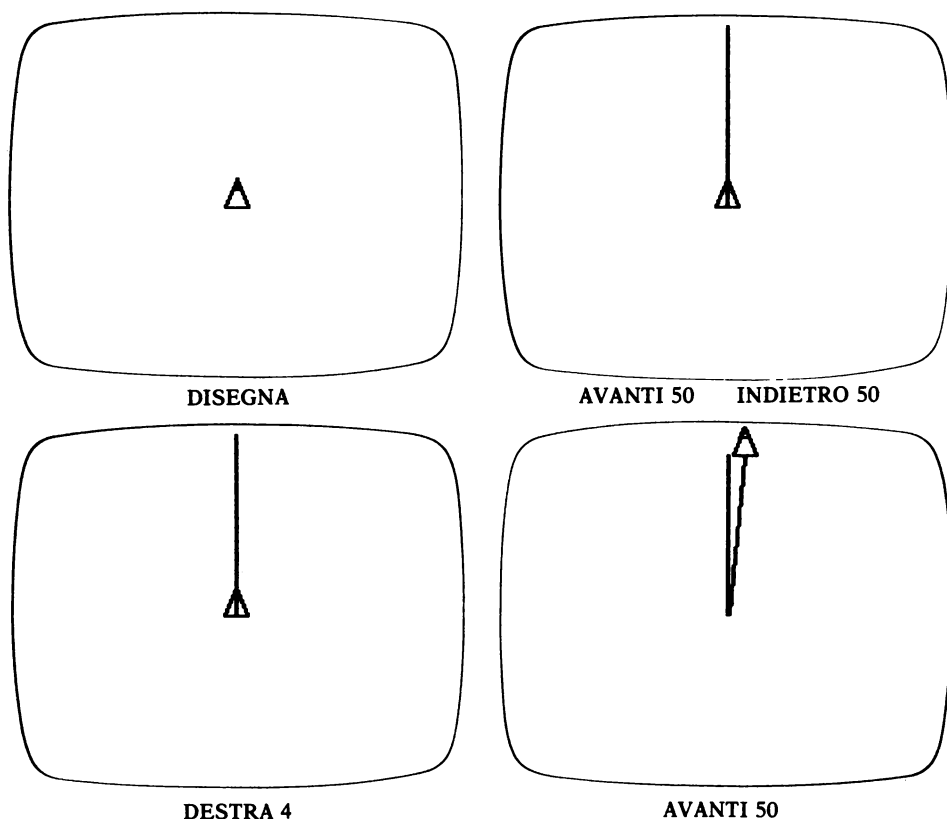
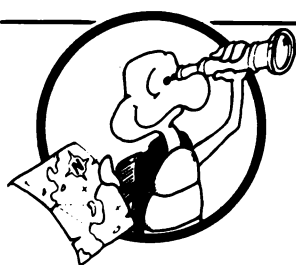



Figura 2.15 La tartaruga non si sposta al comando DESTRA 4, ma la linea viene poi tracciata nella giusta direzione

to l'apparecchio TV o il monitor. Il comando ASCOL richiede di specificare il codice del colore scelto, da 0 a 15.

ASCOL 0 traccia linee nere
ASCOL 1 traccia linee bianche
ASCOL 2 traccia linee rosse
ASCOL 3 traccia linee blu
ASCOL 4 traccia linee viola
ASCOL 5 traccia linee verdi
ASCOL 6 traccia linee blu
ASCOL 7 traccia linee gialle
ASCOL 8 traccia linee arancioni
ASCOL 9 traccia linee marroni

ASCOL 10 traccia linee rosso chiaro
ASCOL 11 traccia linee grigio scuro
ASCOL 12 traccia linee grigio medio
ASCOL 13 traccia linee verde chiaro
ASCOL 14 traccia linee azzurre
ASCOL 15 traccia linee grigio chiaro

Si possono inoltre indicare 16 diversi colori di sfondo, scegliendoli fra quelli appena citati, con il comando COLORESFONDO (oppure CS) che va sempre associato a un codice di colore, da 0 a 15.

ESPLORAZIONE

Ecco qualche idea per esplorare il mondo dei colori con il Logo:

- Battere il comando ASCOL 5 e disegnare poi con la tartaruga un quadrato verde. Cambiare colore e disegnare, accanto al primo, un secondo quadrato di diversa dimensione.
- Battere il comando DISEGNA per ripulire lo schermo e cambiare lo sfondo, con il comando COLORESFONDO 2. Disegnare varie figure su questo sfondo, in bianco, nero e viola (con i comandi ASCOL 0, ASCOL 1 e ASCOL 4). Che cosa succede se si disegna sullo sfondo COLORESFONDO 2 con il colore ASCOL 2?
- Tracciare una figura nera su sfondo bianco.
- Cambiare il colore di sfondo dopo aver tracciato alcune figure.
- Tracciare una figura multicolore. Qui sotto sono indicati i comandi per tracciare un quadrato con lati in colore nero, verde, viola e arancione, su sfondo bianco:

```
COLORESFONDO 1
ASCOL 0
AVANTI 50
DESTRA 90
ASCOL 5
AVANTI 50
DESTRA 90
```

```
ASCOL 4
AVANTI 50
DESTRA 90
ASCOL 8
AVANTI 50
```

TRABOCCHETTO

Per tracciare belle linee brillanti, bisogna ricorrere al comando DUECOLORI, che consente anche di tracciare linee di diversi colori, una accanto all'altra.
Il comando UNCOLORE, invece, fa tracciare linee sottili. Non sempre però i colori corrispondono alla descrizione.
Sia DUECOLORI che UNCOLORE ripuliscono lo schermo e ripristinano lo sfondo e il colore normale del pennino della tartaruga.

2.5 Cerchi e archi

Il Logo non ha comandi incorporati per tracciare cerchi o archi di cerchio; sono necessarie per questo delle procedure, contenute nel disco di utilità. Noi però costruiremo le nostre procedure (sono nell'Appendice A). Chiedete a un amico più esperto o a un grande di ricopiare le procedure dell'Appendice A, altrimenti non rimane che saltare questa parte, per ritornarvi dopo la lettura del Capitolo 4, dove si impara a copiare da soli le procedure scritte in appendice e a registrarle nel proprio disco di servizio.
Se si dispone delle procedure su disco, si inserisce questo nel drive e si batte:

RECUPERA "CERCHI (e RETURN)

TRABOCCHETTO

Per ottenere il carattere ", bisogna battere la combinazione di tasti SHIFT-2.
Il carattere " serve solo prima del nome del file (nella Logolingua, file è il termine generale per indicare ciò che è registrato sul disco sotto un certo nome, che in questo caso è CERCHI).

Dopo aver battuto il comando RECUPERA "CERCHI, si sente il drive ronzare come quando viene caricato in memoria il Logo. Dopo qualche secondo, il Logo scrive sullo schermo i nomi delle sei procedure ora disponibili, come si vede qui di seguito:

```
RECUPERA "CERCHI
LA PROCEDURA CERCHIODES È DEFINITA
LA PROCEDURA ARCOSIN È DEFINITA
LA PROCEDURA ARCODES È DEFINITA
LA PROCEDURA RCP È DEFINITA
LA PROCEDURA LCP È DEFINITA
LA PROCEDURA CERCHIOSIN È DEFINITA
```

In questo modo, il Logo fa sapere di avere ora a disposizione un insieme di procedure per tracciare cerchi e archi di cerchio. CERCHIODES e CERCHIOSIN disegnano cerchi. ARCODES e ARCOSIN disegnano archi di cerchio di 90 gradi (chiamati anche quadranti). Ciascuno dei nuovi comandi richiede l'assegnazione di un numero, che rappresenta il raggio del cerchio considerato. (Un arco fa sempre parte di un cerchio. Il raggio del cerchio è la distanza dei punti della circonferenza dal centro.)
Ecco qualche esempio:

```
CERCHIODES 20
CERSHIOSIN 10
ARCOSIN 20
ARCODES 10
```

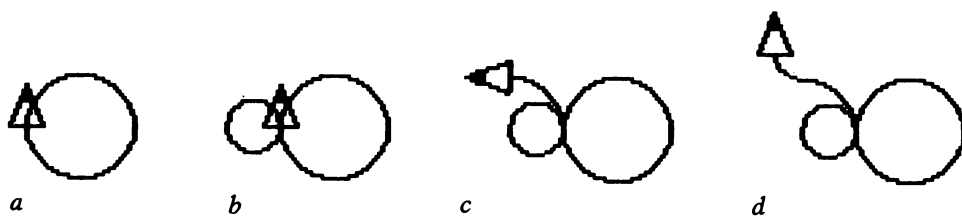
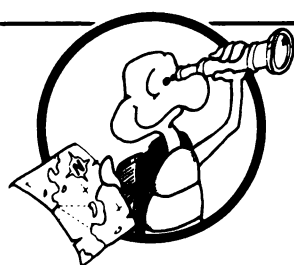



Figura 2.16 Disegni tracciati con i comandi: CERCHIODES 20 (*a*); CERCHIOSIN 10 (*b*); ARCOSIN 20 (*c*); ARCODES 10 (*d*)

Ecco che cosa si può fare con archi e cerchi (il resto si vedrà nel Capitolo 5):

- Tracciare cerchi di diversa grandezza, facendoli espandere verso destra.

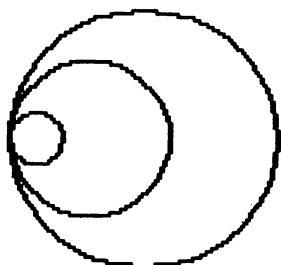



Figura 2.17 Cerchi di diversa grandezza, tracciati con CERCHIODES

- Tracciare ancora cerchi di diversa grandezza, facendoli però espandere verso sinistra.
- Combinare i cerchi appena tracciati, in modo da formare una "farfalla".

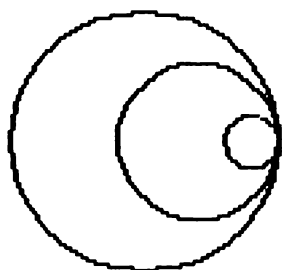



Figura 2.18 Cerchi di diversa grandezza, tracciati con CERCHIOSIN

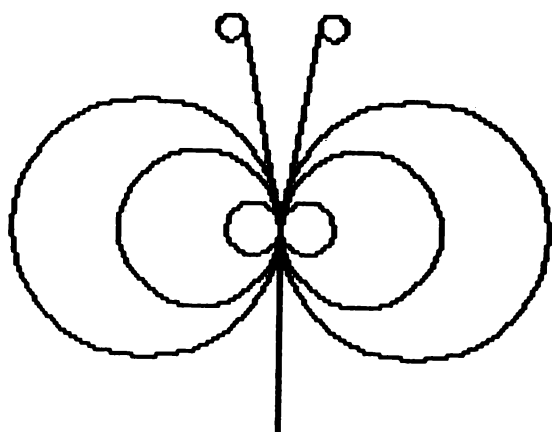



Figura 2.19 Figura a forma di farfalla, costruita con cerchi

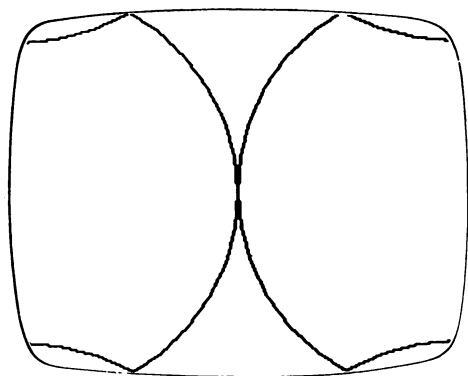



Figura 2.20 Ecco un cerchio tanto grande da avvolgere lo schermo e congiungersi con se stesso

Ecco ancora qualche spunto:

- È interessante vedere che cosa succede quando i cerchi diventano grandissimi:

  CERCHIODES 50
  CERCHIODES 100
  CERCHIODES 200
  CERCHIODES 500
  e così via.....

- Che numero bisogna assegnare per tracciare un cerchio che avvolge lo schermo e si riunisce al centro, come in Figura 2.20?
- Si riesce a disegnare il motivo di Figura 2.21 con i comandi ARCODES e ARCOSIN?

Figura 2.21 Motivo ondeggiante costruito con ARCODES e ARCOSIN

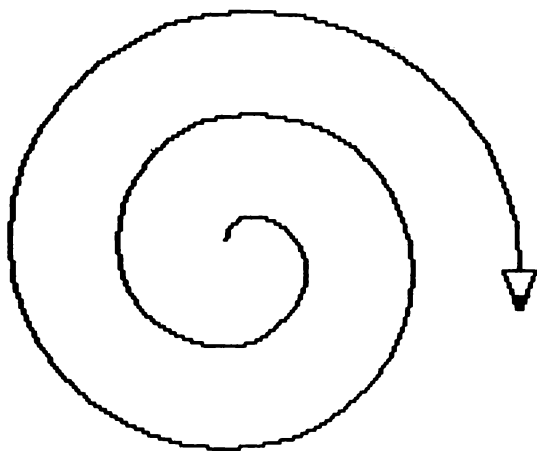



Figura 2.22 Spirale tracciata con ARCODES chiamato più volte, e con raggio sempre più grande

- Si riesce a tracciare una **spirale**, come quella di Figura 2.22, assegnando via via raggi sempre **più grandi**?

SUGGERIMENTO

I cerchi e gli archi di cerchio sono ideali per considerare la simmetria delle figure. Il Logo realizza disegni simmetrici semplicemente eseguendo i comandi nella versione DESTRA e nella versione SINISTRA, e combinando poi i risultati grafici. È possibile tracciare altre figure, oltre al cerchio, con simmetria destra o sinistra: si vedranno più avanti applicazioni di questo tipo, con i comandi QUADRATODES e QUADRATOSIN, STELLADES e STELLASIN, ed altri ancora.

2.6 Altri comandi di manovra della tartaruga

Ecco alcuni altri comandi, che tornano utili in molte occasioni e che compariranno nelle costruzioni presentate più avanti nel libro.

NASTARTA (o NT) e MOSTARTA (o MT) servono per eliminare (o per ripristinare) la visualizzazione della tartaruga, senza spostarla da dove si trova. A volte si fa sparire la tartaruga per ottenere un migliore effetto grafico.

a

b

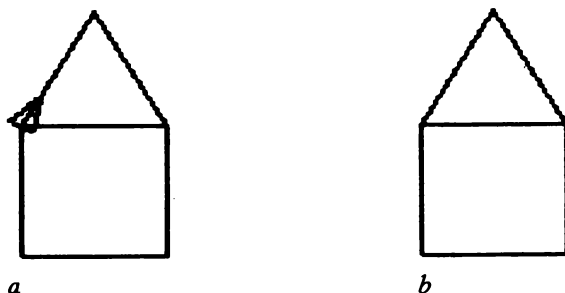



Figura 2.23 Disegno con la tartaruga visualizzata (*a*) e non (*b*)

Se si elimina la visualizzazione della tartaruga sin dall'inizio, si riesce a tracciare i disegni un poco più in fretta: ciò torna utile per disegni complessi, in cui magari si intersecano molte linee, cerchi, e figure di vario genere. Ecco un esempio:

```
DISEGNA
CERCHIODES 50
DISEGNA
NASTARTA
CERCHIODES 50
MOSTARTA
```

Si nota la differenza di velocità?

Il comando MURO obbliga la tartaruga a rimanere all'interno dello schermo. Se allora si cerca di oltrepassare un bordo, il Logo protesta, come in questo esempio:

```
DISEGNA
MURO
AVANTI 200
```

Il comando ATTORNO ripristina la situazione normale. Ecco un esempio da provare:

```
MURO
AVANTI 200
ATTORNO
AVANTI 200
```

Il comando PULISCISCHERMO (ovvero PS) ripulisce completamente lo schermo, senza però spostare la tartaruga. Per vedere la differenza fra PULISCISCHERMO e DISEGNA, basta provare questo esempio:

```
DISEGNA
DESTRA 30
AVANTI 20
PULISCISCHERMO
AVANTI 20
```

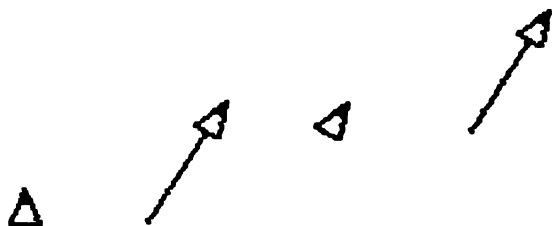



Figura 2.24 Prova dell'effetto **del comando** PULISCISCHERMO

Il comando TANA riporta la tartaruga al centro dello schermo, rivolta in alto. La combinazione di TANA e di PULISCISCHERMO equivale al comando DISEGNA, come si vede in questo esempio:

```
DISEGNA
AVANTI 50
TANA
PULISCISCHERMO
```

Se si prova a invertire l'ordine dei comandi, scambiando TANA con PULISCISCHERMO, si ottiene un effetto diverso, come si vede provando questo esempio:

```
DISEGNA
AVANTI 50
PULISCISCHERMO
TANA
```

Come mai rimane una linea sullo schermo?

Ecco infine alcuni comandi che riguardano l'aspetto generale dello schermo:

GRASCHERMO (oppure f5) rende visibile l'intero schermo, ma esclude dalla visualizzazione tutti i comandi dati al Logo e i messaggi con cui esso risponde.

DISCHERMO (oppure f3) ripristina la suddivisione normale dello schermo. Così si vedono cinque righe dedicate alle scritte, ma viene nascosta una parte del campo d'azione della tartaruga.

TESCHERMO (oppure f1) dedica tutto lo schermo alle scritte, coprendo l'intero campo d'azione della tartaruga. Per rivedere la tartaruga, basta battere f5 oppure f3.

Si può provare ora questo esempio riassuntivo:

```
DISEGNA
SULAPENNA
SINISTRA 90
AVANTI 100
GIULAPENNA
DESTRA 90
CERCHIODES 100
```

GRASCHERMO
DISCHERMO
TESCHERMO
DISCHERMO

I comandi in forma abbreviata f5, f3 e f1 possono essere dati in qualunque momento, senza dover concludere con il tasto RETURN. Se invece si sceglie la forma estesa dei comandi (GRASCHERMO, DISCHERMO o TESCHERMO) bisogna concludere, come al solito, con la battuta del tasto RETURN.

TRABOCCHETTO

I comandi ESCIDIS e TESCHERMO sembrano, a prima vista, avere lo stesso effetto. Ci si accorge della differenza quando si richiama un comando di manovra della tartaruga: se ad esempio si richiama DISEGNA (o un qualunque altro comando di manovra) quando è attivo il comando ESCIDIS, si ottiene immediatamente lo schermo suddiviso in due parti, mentre ciò non accade se è attivo il comando TESCHERMO. Il Logo esegue comunque il comando grafico, ma il campo d'azione della tartaruga rimane nascosto, finché non si batte f5, oppure f3.

SUGGERIMENTO

Il comando TANA sposta sempre e comunque la tartaruga al centro dello schermo. In questo, il comando è diverso da AVANTI, INDIETRO, DESTRA e SINISTRA, che realizzano spostamenti relativi, partendo dalla posizione

in cui al momento si trova la tartaruga.
Il Logo ha altri comandi grafici che, come TANA, pongono la tartaruga in una certa posizione dello schermo, senza riguardo per la sua posizione attuale. I comandi VAX, VAY e VAXY collocano la tartaruga nella posizione corrispondente alla coordinata cartesiana X e/o Y specificata. La posizione attuale della tartaruga può essere visualizzata con i comandi XCOL e YCOL.
Il comando ASDIR fissa l'orientamento della tartaruga nella direzione specificata (mentre DIREZIONE visualizza la direzione in cui la tartaruga è orientata al momento).
Questi comandi sono utili in molte occasioni, ma sono stati qui volutamente trascurati per concentrare l'attenzione su quelli più fecondi dal punto di vista dell'apprendimento, che sono quelli di orientamento relativo, rispetto alla posizione attuale. I comandi di collocazione in coordinate assolute sono presentati quando tornano utili, di volta in volta. Ad esempio, DIREZIONE viene presentato nel Capitolo 7, là dove si parla dei comandi condizionati. Invece ASDIR e VAXY sono presentati, con il sistema di coordinate cartesiane, nel Capitolo 10, dove servono per il gioco TIRO.A.SEGNO.

3

Due applicazioni della tartaruga

Qui si presenta un gioco, chiamato TIRO.A.SEGNO, e uno strumento grafico, che si chiama SCHIZZO. Entrambi offrono l'occasione di utilizzare la tartaruga in modo costruttivo e di comprenderne bene le possibilità di impiego. TIRO.A.SEGNO, in particolare, aiuta a capire angoli e distanze, mentre SCHIZZO consente di tracciare con grande facilità i disegni più vari e interessanti. Più avanti, in questo stesso libro, si vedrà come sono scritti (in Logo) i programmi TIRO.A.SEGNO e SCHIZZO, che sono per la verità piuttosto complicati: in questo capitolo ci si limita invece ad usarli, caricandoli semplicemente in memoria dopo averli copiati dall'Appendice A.
In questo capitolo non vengono usati nuovi comandi.

SUGGERIMENTO

TIRO.A.SEGNO e SCHIZZO offrono l'occasione di un'attività stimolante anche ai principianti, che in tal modo possono prendere confidenza con la manovra della tartaruga. Sono un esempio di quelli che Seymour Papert chiama microcosmi educativi (nel suo libro *Mindstorms*).

Questi microcosmi sono ambienti ideali per chi impara, in quanto l'addestramento avviene in modo interessante e vivo, con l'esercizio pratico di concetti essenziali. L'abilità di un buon insegnante di Logo sta anche nel saper ideare e realizzare microcosmi educativi, da proporre a chi impara (o almeno nel saper indirizzare l'allievo verso la creazione autonoma dei propri microcosmi). Non è detto che per questo vadano sempre scritti dei programmi per il computer.
Molti esempi di tale modo di imparare sono già stati presentati nelle parti dedicate all'esplorazione autonoma dei capitoli precedenti. Se si intende procedere su questa strada, occorre però una copia su disco delle procedure riportate in Appendice A. Prima di copiare le procedure, bisogna però leggere i Capitoli 10 e 11, che le descrivono.

3.1 TIRO.A.SEGNO, gioco interattivo con tartaruga

Prima di giocare a TIRO.A.SEGNO, conviene ripulire la memoria del computer, con questo comando:

CIAO (e RETURN)

Se le avete già battute, potete leggere le procedure dal disco, caricando il file TIRO.A.SEGNO. Per farlo si inserisce il disco nel drive, battendo poi il seguente comando:

RECUPERA "TIRO.A.SEGNO (e RETURN)

A questo punto il drive si mette a ronzare e a ticchettare per un po', fino a che non compare sullo schermo un lungo elenco, con i nomi delle procedure caricate:

```
?RECUPERA "TIRO.A.SEGNO
LA PROCEDURA CERCHIETTO È DEFINITA
LA PROCEDURA ASPETTA È DEFINITA
LA PROCEDURA MANCATO È DEFINITA
LA PROCEDURA COLPITO È DEFINITA
LA PROCEDURA DISTANZA È DEFINITA
LA PROCEDURA LEGGINUMERO È DEFINITA
LA PROCEDURA TARTARUGAVAI È DEFINITA
LA PROCEDURA TRACCIABERSAGLIO È DEFINITA
LA PROCEDURA INCOMINCIA È DEFINITA
LA PROCEDURA INIZIADATI È DEFINITA
LA PROCEDURA VIA È DEFINITA
```

LA PROCEDURA RCP È DEFINITA
LA PROCEDURA TIRO È DEFINITA
?

Ecco come si gioca:

1. Battere:

    VIA (e RETURN)

   Il computer presenta un bersaglio, in un punto qualunque dello schermo (scelto a caso, in modo automatico). Ogni volta il punto cambia. La tartaruga viene anch'essa collocata sullo schermo, in un altro punto anch'esso scelto a caso.
2. Mirare al bersaglio, puntandovi la tartaruga (con i comandi di rotazione, SINISTRA e DESTRA).
3. Quando si è mirato, battere:

    TIRO (e RETURN)

4. Il computer visualizza la domanda A CHE DISTANZA?. Si risponde battendo un numero, che indica la distanza di tiro. Il computer sposta allora la tartaruga e indica se il bersaglio è colpito o no. Se il bersaglio è mancato, la tartaruga ritorna al punto di partenza e si può provare un altro colpo. Questo è tutto.

Ecco una mia partita. Ho incominciato battendo:

VIA (e RETURN)

Figura 3.1 Inizio di una partita con TIRO.A.SEGNO

Il computer ha collocato il bersaglio e la tartaruga, come si vede in Figura 3.1.
Ho puntato poi la tartaruga, stimando una rotazione di 100 gradi, battendo il comando:

DESTRA 100

Figura 3.2 Tartaruga ruotata di 100 gradi

Poi ho preferito correggere la mira, ruotando di altri 10 gradi la tartaruga, sempre verso destra, con il comando:

DESTRA 10

Figura 3.3 Tartaruga ruotata di altri 10 gradi

Sono passato quindi alla distanza, battendo:

TIRO

Il computer ha risposto:

A CHE DISTANZA?

Ha scelto la distanza di 100 passi, battendo:

100

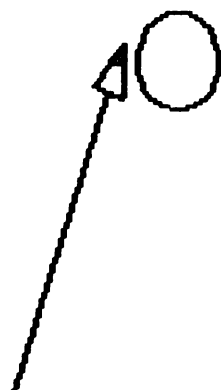



Figura 3.4 Risultato del primo colpo

Ho mancato il bersaglio, anche se di poco.
La tartaruga è allora ritornata al punto di partenza, lasciando sullo schermo la traccia del primo colpo. Ho ripreso la mira: ora però avevo un riferimento, la traccia del mio primo colpo, che era stato un po' corto e un po' troppo a sinistra. Dovevo allora aumentare la rotazione della tartaruga, che era stata di 110 (prima 100 e poi altri 10). Ho battuto quindi:

DESTRA 120

Dovevo anche allungare un po' il tiro, rispetto al primo colpo, e così ho provato la distanza 125:

TIRO
A CHE DISTANZA?
125

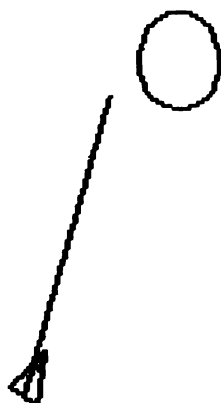



Figura 3.5 Tartaruga ruotata di 120 gradi rispetto all'orientamento di partenza

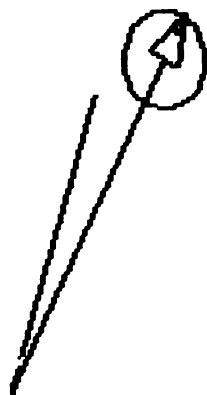



Figura 3.6 Centro!

Il messaggio finale è stato:

CONGRATULAZIONI! BERSAGLIO CENTRATO!
CI SONO VOLUTI SOLO 2 COLPI

Questo è TIRO.A.SEGNO: ora lo potete provare.

SUGGERIMENTO

TIRO.A.SEGNO è una buona occasione per fare pratica con la manovra della tartaruga e si rivela utile soprattutto per chi ha difficoltà a valutare ad occhio distanze e angoli (ad esempio, i bambini piccoli). Le ricerche condotte al MIT sull'addestramento al Logo hanno dimostrato che, per manovrare la tartaruga con successo, occorre avere acquisito una buona manualità è un discreto colpo d'occhio, che TIRO.A.SEGNO aiuta a sviluppare.
Insegnando il Logo non bisogna mai trascurare la manualità pratica dei comandi, che può a volte essere di ostacolo per l'allievo: occorre sempre fare attenzione a questo tipo di problema, e favorire la pratica delle abilità manuali richieste dalla manovra della tartaruga.
TIRO.A.SEGNO è un esempio di ausilio didattico computerizzato, che può essere facilmente modificato dal genitore, dall'insegnante, e in genere da chiunque conosca il Logo. Per un piccolo allievo, può essere importante vedere qualcuno che modifica il Logo, perché così capisce che si tratta di uno strumento controllabile (anche se solo da una persona esperta).
Si può immaginare un'attività molto utile, adatta per una coppia di allievi, uno agli inizi e l'altro già esperto, in cui il gioco TIRO.A.SEGNO viene modificato man mano che il principiante lo impara. Alcune indicazioni sulla modifica del gioco compaiono nel Capitolo 10.
TIRO.A.SEGNO può essere giocato da un gruppo di piccoli allievi anche all'aperto, simulandone le varie parti con un movimento reale (chi impersona la tartaruga può fare all'inizio come a mosca cieca, per rappresentare la casualità della posizione e dell'orientamento iniziale; gli altri possono poi dirigere questa tartaruga gridando semplicemente i comandi, invece di batterli alla tastiera).
Come si è già osservato nel paragrafo 2.3, è essenziale in queste esercitazioni pratiche fare molta chiarezza sulle regole del gioco. Può darsi, comunque, che ci voglia un po' di esercizio: ciò che si dovrebbe cercare non è però la riproduzione esatta del gioco al computer, ma l'arricchimento di questo con nuovi stimoli ed idee (e viceversa, naturalmente).

3.2 SCHIZZO, ovvero come disegnare comodamente

Con SCHIZZO il disegno con la tartaruga diventa più comodo e immediato, perché basta battere un solo tasto per manovra, senza nemmeno dover battere il tasto RETURN. È anche possibile dare un nome al disegno fatto, e richiamarlo in seguito, quando si vuole.

Prima di usare SCHIZZO, bisogna ripulire la memoria del computer, battendo CIAO. Si recupera poi dal disco il file chiamato "SCHIZZO (o, se non l'avete già fatto, copiate su disco le procedure dell'Appendice A). Ecco i comandi:

CIAO (e RETURN)
RECUPERA "SCHIZZO (e RETURN)

Il computer presenta l'elenco dei nomi delle procedure che fanno parte del file "SCHIZZO:

```
?RECUPERA "SCHIZZO
LA PROCEDURA AGGLETTERA È DEFINITA
LA PROCEDURA LEGGITASTO È DEFINITA
LA PROCEDURA TERMINA È DEFINITA
LA PROCEDURA RICOMANDO È DEFINITA
LA PROCEDURA COMANDO È DEFINITA
LA PROCEDURA VIA È DEFINITA
LA PROCEDURA M È DEFINITA
LA PROCEDURA RD È DEFINITA
LA PROCEDURA SCHIZZO È DEFINITA
?
```

Ecco come si usa SCHIZZO. Si batte:

M (e RETURN)

Per disegnare con la tartaruga, sono allora a disposizione i seguenti tasti (che non vanno seguiti dalla battuta del tasto RETURN):

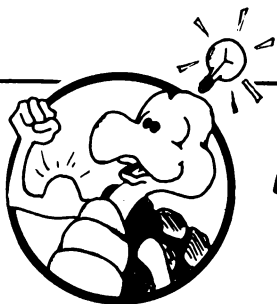
A sposta la tartaruga in avanti di 20 passi
I sposta la tartaruga all'indietro di 20 passi
D ruota la tartaruga verso destra di 30 gradi
S ruota la tartaruga verso sinistra di 30 gradi

Per terminare un disegno e assegnare alla figura realizzata un nome (se si vuole), si batte:

Z

Il computer risponde allora:

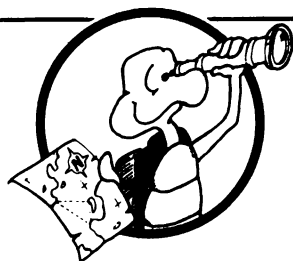
```
COME CHIAMI IL DISEGNO?
(UNA SOLA PAROLA)
SE LO VUOI BUTTARE, BATTI SOLO IL TASTO RETURN
```

Se si batte un nome (come si vuole, di una sola parola), alla battuta del tasto RETURN il computer lo memorizza. Si può provare a tracciare uno schizzo qualunque, battendo alla fine il tasto Z e assegnando il nome SGORBIO.

Il nome da assegnare al disegno è completamente libero (basta che sia di una sola parola): invece di SGORBIO, si può scegliere ANTONIO, BELLO, o KIYU899Z2, se si vuole.

Per ritrovare il disegno chiamato SGORBIO, basta battere un semplice comando:

RD :SGORBIO (e RETURN)

TRABOCCHETTO

I due punti (:) sono molto importanti nel Logo. Il comando RD :SGORBIO indica al computer di ridisegnare sullo schermo il disegno chiamato SGORBIO. Se ci si dimentica di battere i due punti appena prima del nome, o si lascia uno spazio bianco (come ad esempio in : SGORBIO), non si ottiene nulla e il Logo protesta.

ESPLORAZIONE

SCHIZZO serve a disegnare praticamente qualunque figura venga in mente. Ecco qualche idea, tanto per incominciare:

- Tracciare forme semplici, come triangoli, quadrati, o cerchi.

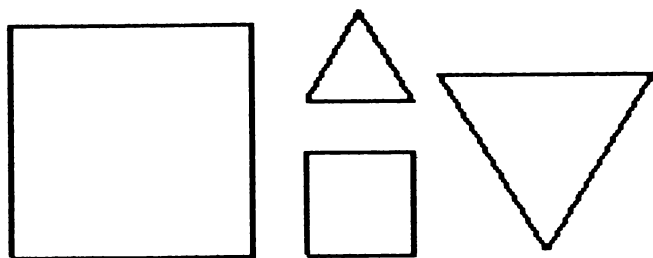



Figura 3.7 Quadrati e triangoli tracciati con SCHIZZO

- Ripetere una figura semplice per più volte, ruotando man mano la tartaruga, fino a ritornare all'orientamento di partenza.

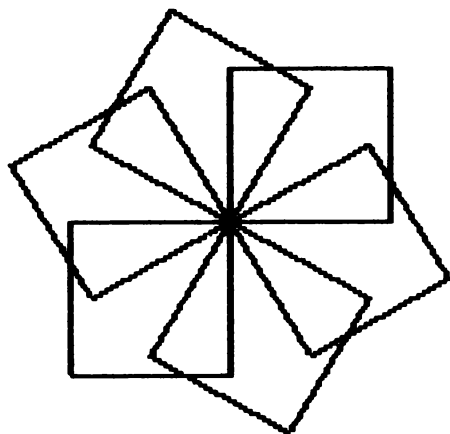



Figura 3.8 Disegno ottenuto ripetendo più volte un quadrato, e ruotando man mano la tartaruga

- Ripetere una figura in varie parti dello schermo, ricorrendo anche ai comandi SULAPENNA e GIULAPENNA. Si possono anche disporre le figure in modo da formare lo schizzo di un viso.

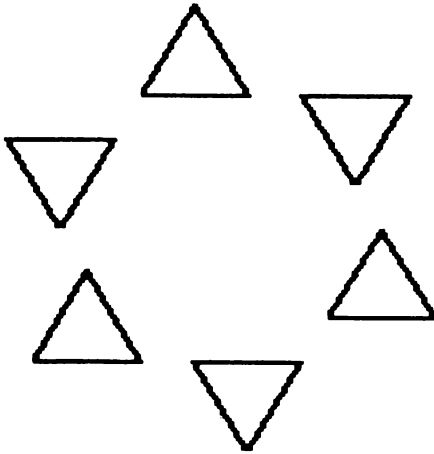



Figura 3.9 Figure disposte in parti diverse dello schermo

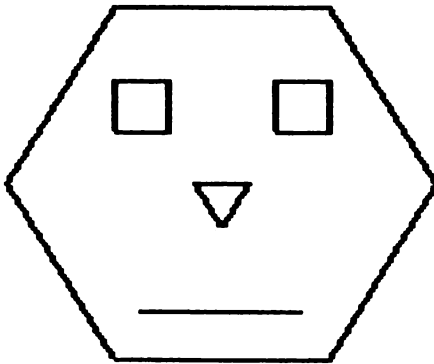



Figura 3.10 Figure disposte a formare lo schizzo di un viso

- Tracciare una figura a caso, senza pensarci. Assegnare quindi un nome alla figura e ripeterla più volte, in nuove parti dello schermo, fino ad ottenere un motivo che, a volte, può riuscire davvero intressante.
- Scegliere qualcuno dei disegni proposti nei Capitoli 5 e 6, e cercare di realizzarli con SCHIZZO.

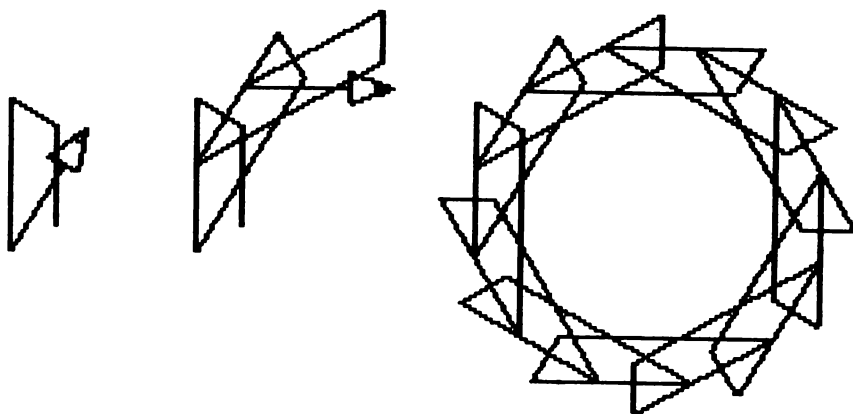



Figura 3.11 Motivo costruito ripetendo più volte una figura tracciata a caso

SUGGERIMENTO

SCHIZZO è un microcosmo comprendente, anche se in modo limitato, molti dei concetti più importanti per l'apprendimento del Logo:

- La manovra della tartaruga (avanzamento e rotazione).
- La denominazione di una figura e la sua riproduzione.
- La costruzione di un disegno complesso per mezzo di blocchi elementari.
- La costruzione di un disegno ripetitivo in forma chiusa.

SCHIZZO viene presentato qui per tre motivi. In primo luogo, è un buon ambiente di apprendimento per piccoli allievi che possono trovare ancora troppo complessa la struttura dei comandi del Logo. Del resto, anche i nomi stessi delle procedure (M per SCHIZZO e RD per RIDISEGNA) sono stati scelti in modo da richiedere il minimo di abilità manuale alla tastiera. Questo fatto può contare parecchio in gruppi dove sono presenti allievi di età e di esperienza diverse, perché così anche i più piccoli e inesperti possono partecipare con i più grandi a molte delle loro attività con il Logo.
Con SCHIZZO è inoltre facile sperimentare alcuni dei tipi di disegno da realizzare con la tartaruga, che verranno discussi con maggior dettaglio nei Capitoli 5 e 6.

Con SCHIZZO si ha inoltre un esempio di progetto di media complessità, realizzabile con il Logo (nel Capitolo 11 si può trovare una descrizione esauriente, con suggerimenti di estensione e di modifica).
La versione estesa di SCHIZZO descritta nel Capitolo 11 è particolarmente adatta all'impiego da parte di allievi molto piccoli (ad esempio, con età da tre a sei anni): in alternativa, si può anche considerare il programma INSTANT, descritto nel libro *Apple Logo*, di Harold Abelson.
SCHIZZO offre un'eccellente occasione di collaborazione fra due allievi di diversa esperienza, o anche fra un insegnante e un allievo: il piccolo inesperto può infatti proporre egli stesso estensioni e modifiche, che può vedere realizzate dal più esperto (con l'aiuto delle spiegazioni date nel Capitolo 11).
Bisogna anche dire che SCHIZZO non intende in alcun modo rappresentare un ambiente di apprendimento definitivo e perfetto: un simile ambiente non esiste, se non in quanto viene costruito di volta in volta, da grandi e piccoli, tenendo conto delle esigenze e delle circostanze contingenti. Inoltre, si è voluta limitare la complessità di SCHIZZO, per poter offrire con esso anche un esempio tipico di programma di media grandezza, da presentare come esempio di inizio della creazione di un microcosmo educativo.
SCHIZZO va bene per chi incomincia e desidera fare subito qualcosa di interessante, ma non può rappresentare il traguardo finale dell'apprendimento, e tanto meno lo strumento applicativo ideale. Ad un certo punto, tutti dovrebbero staccarsi da SCHIZZO e proseguire con il Logo, proprio come si passa dall'antipasto al piatto forte.

4

Come insegnare al computer

È possibile insegnare al computer comandi nuovi (detti anche procedure). In questo capitolo si vede come insegnare al computer nuovi comandi, con l'aiuto del Logo (che ha uno strumento già pronto per comporre testi sullo schermo). Si vede anche in che modo si registrano su disco le procedure (salvandole, come anche si dice, su un file), come si registrano sul proprio disco di servizio le figure costruite con il Logo, e come si ottengono copie su carta di ciò che è rappresentato sullo schermo (figure, o procedure), per mezzo della stampante.
D'ora in poi, in effetti, si assegneranno sempre nuovi comandi al computer, registrando su disco le procedure che li descrivono. Proseguendo nella lettura del libro, può anche darsi che si debba ritornare a questo capitolo per qualche chiarimento o per approfondire un argomento nuovo.
Durante la lettura, occorre avere a disposizione un disco di servizio e il diario del Logo. Per creare un disco di servizio, se già non se ne ha uno, basta seguire le indicazioni dell'Appendice B.

Nuovi comandi presentati nel capitolo:

Comando	**Abbreviazione**	**Esempi**
PER		PER QUADRATO
FINE	END	
EDITA	E	EDITA QUADRATO, E QUADRATO
FAIVEDERE	FV	FAIVEDERE QUADRATO, FV QUADRATO
CANCELLA	CA	CANCELLA QUADRATO, CA QUADRATO CANCELLA TUTTO

Comando	Abbreviazione	Esempi
CONSERVA		CONSERVA "CERCHIO
RECUPERA		RECUPERA "CERCHIO
CATALOGO		
CANCEARC		CANCEARC "ROBAVECCHIA
CONSERDIS		CONSERDIS "QUADRATO
RECUPDIS		RECUPDIS "QUADRATO
CANCEDIS		CANCEDIS "QUADRATO
STAMPANTE		
NOSTAMPANTE		

4.1 Come insegnare al computer a fare QUADRATO

Il lettore sa già, probabilmente, tracciare sullo schermo un quadrato, con la tartaruga (se no, dovrebbe rileggere il Capitolo 2). Se però non si trova un modo veloce di far eseguire semplici compiti come il disegno di un quadrato, l'uso del computer sarebbe veramente noioso. Il Logo ci aiuta però ad insegnare nuovi comandi al computer, che così diventa più svelto e potente. In altri termini, il Logo ci facilita di molto la cosiddetta programmazione del computer. Vediamo allora come si fa ad insegnare al computer il modo di tracciare un quadrato.
Bisogna prima spiegare al computer che cosa significa QUADRATO, battendo queste istruzioni:

```
PER QUADRATO
AVANTI  50
DESTRA  90
AVANTI  50
DESTRA  90
AVANTI  50
DESTRA  90
AVANTI  50
DESTRA  90
FINE
```

La prima riga, PER QUADRATO, dice al computer che viene assegnato un nuovo comando, chiamato QUADRATO.
Le righe seguenti sono le istruzioni che il computer deve seguire per fare QUADRATO.
L'ultima parola, FINE, segnala al computer che le istruzioni del comando QUADRATO sono terminate.

Seguiamo allora l'esempio passo passo: si batte una riga alla volta, leggendo con attenzione i commenti.
Prima istruzione:

PER QUADRATO (e RETURN)

Quando si batte questa istruzione, il computer ripulisce lo schermo e si predispone alla composizione (altrimenti detta editing, nel gergo degli addetti ai lavori); l'aspetto dello schermo cambia: tutto viene visualizzato ora con scrittura inversa, cioè bianco su nero, e compare in fondo la scritta:

DEFINISCI CON CTRL-C, ANNULLI CON CTRL-G

Con il computer così predisposto, si è liberi di battere qualunque testo, che viene immediatamente visualizzato sullo schermo; il computer non fa altro che questo, fino a che non si conclude la fase di composizione.
Si prosegue battendo, uno alla volta, i comandi riportati qui di seguito:

```
PER QUADRATO   (e RETURN)
AVANTI 50      (e RETURN)
DESTRA 90      (e RETURN)
AVANTI 50      (e RETURN)
DESTRA 90      (e RETURN)
AVANTI 50      (e RETURN)
DESTRA 90      (e RETURN)
AVANTI 50      (e RETURN)
DESTRA 90      (e RETURN)
FINE           (e RETURN)
```

Se si fanno errori di battitura, non c'è problema: è facile correggerli con i tasti segnati dalle freccette e col tasto INST/DEL.
Dopo aver battuto FINE, si deve ancora terminare la fase di composizione: si batte la coppia di tasti CTRL-C (tenendo premuto il tasto CTRL mentre si batte il tasto C), oppure anche il tasto RUN/STOP. Il computer scrive allora:

UN MOMENTO PREGO
LA PROCEDURA QUADRATO È DEFINITA

TRABOCCHETTO

Succede spesso che ci si dimentica di battere i tasti CTRL-C quando si è terminata la battitura di una procedura nuova. Se non si compie questa operazione, tutti i comandi vengono ignorati, perché il computer li esegue solo quando è predisposto a farlo, e ciò si ottiene con i tasti CTRL-C oppure con il tasto RUN/STOP.

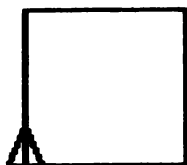
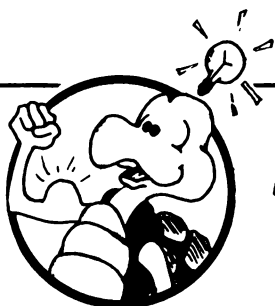



Figura 4.1 Ecco che cosa si ottiene con il nuovo comando QUADRATO

Quando si batte la coppia di tasti CTRL-C, oppure il tasto RUN/STOP, si conclude la fase di composizione e si ripristina lo stato operativo normale del computer, in cui i comandi vengono eseguiti immediatamente, non appena sono battuti alla tastiera. Se, ad esempio, si batte ora il comando QUADRATO, si ottiene sullo schermo il disegno di Figura 4.1.
Se la nuova procedura non fa quello che ci si aspetta, la si può correggere. Si batte allora:

EDITA QUADRATO (e RETURN)

Così facendo, si ritorna alla fase di composizione, e si possono correggere gli errori con i tasti contrassegnati dalle freccette e col tasto INST/DEL. Alla fine, si batte la coppia di tasti CTRL-C e si riprova con l'esecuzione del comando QUADRATO.

QUADRATO è ora un comando Logo come tutti gli altri, anche se è stato appena definito. Se si vuole, si può costruire un linguaggio di comando completamente nuovo, definendo tutti i comandi nel modo visto con QUADRATO.
Se si vuole registrare una procedura su disco, si devono seguire le indicazioni del paragrafo 4.3. Nel Capitolo 5 si vede, poi, come usare QUADRATO nel ruolo di procedura di base, con la quale comporre altri nuovi comandi che costruiscono figure più complesse, basate però tutte sul quadrato, come si vede in Figura 4.2.

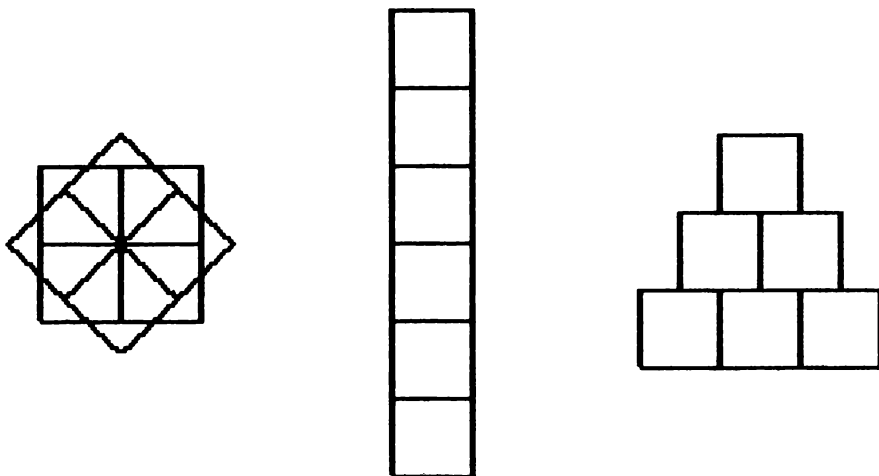



Figura 4.2 Disegni costruiti con l'impiego della procedura QUADRATO

COME DARE UN'OCCHIATA AL LAVORO FATTO

Si può vedere l'elenco delle istruzioni della procedura PER QUADRATO semplicemente battendo:

FV QUADRATO

FV è l'abbreviazione del comando FAIVEDERE. In questo modo, l'elenco viene scritto sullo schermo; se non si riesce a vedere l'intero elenco, si può battere f1, che assegna l'intero schermo alla visualizzazione del testo (alla fine, si ritorna alla normale suddivisione dello schermo battendo f3). Per vedere uttte le procedure, si batte:

FV TUTTO

Con il comando FV TITOLI del Logo si ottiene invece l'elenco dei nomi di tutte le procedure. FV TITOLI è l'abbreviazione di FAIVEDERE TITOLI. Se si vuole cancellare una procedura, ad esempio la procedura QUADRATO, si batte:

CANCELLA QUADRATO

oppure:

CA QUADRATO

Si possono anche cancellare dalla memoria del computer tutte le procedure, battendo:

CANCELLA TUTTO

oppure:

CA TUTTO

TRABOCCHETTO

Bisogna fare molta attenzione. Non bisogna mai battere il comando CANCELLA TUTTO se non si è sicuri di avere registrato su disco le procedure che interessano, o se non si è certi che siano proprio da eliminare tutte quante. Nel paragrafo 4.3 si trovano le indicazioni per registrare su disco le procedure.

4.2 La composizione di testi con il Logo

Il Logo rende facile la scrittura di una procedura nuova, o la modifica di una già esistente. Può darsi che ci voglia un po' di tempo ad abituarsi allo strumento che il Logo mette a disposizione per comporre testi, dato che esso lavora in un modo particolare.

LA COMPOSIZIONE

Ogni attività ha i suoi termini tecnici caratteristici. La composizione di testi non fa eccezione, e fa uso di alcuni termini particolari. Le lettere, i numeri e i simboli battuti alla tastiera e scritti sullo schermo vengono chiamati caratteri. Inoltre, l'insieme dei caratteri scritti sullo schermo forma un testo. Il quadratino lampeggiante sullo schermo è detto cursore, e segnala la posizione in cui verrà scritto il prossimo carattere. Il cursore viene manovrato con due tasti, detti tasti di spostamento, che combinati con il tasto SHIFT lo possono collocare in qualunque punto del testo. Tutte le operazioni di composizione richiedono solo i tasti di spostamento, il tasto RETURN e il tasto INST/DEL.

TASTI DI SPOSTAMENTO DEL CURSORE

⇐CRSR⇒
Sposta il cursore di una posizione verso destra (ha l'autoripetizione, se lo si tiene premuto).

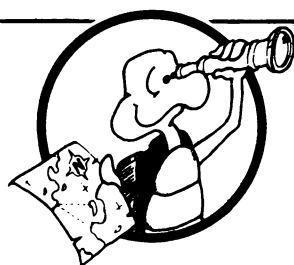
SHIFT ⇐CRSR ⇒
Sposta il cursore verso sinistra di una posizione. Anch'esso ha l'autoripetizione, se lo si tiene premuto. Anche il tasto con la freccia verso sinistra ← sposta il cursore verso sinistra di una posizione.

SHIFT ⇑CRSR ⇓
Sposta il cursore di una riga verso l'alto. È dotato di autoripetizione. Il tasto ↑ ha il medesimo effetto.

⇓CRSR⇑
Sposta il cursore di una riga verso il basso. È dotato di autoripetizione.

TASTI PER LA MODIFICA DEL TESTO

INST/DEL
Cancella il carattere a sinistra del cursore, e fa arretrare questo di una posizione.

RETURN
Crea una nuova riga, collocandovi il cursore all'inizio.

ESPLORAZIONE

Conviene fare un po' di pratica di composizione. Si può battere, ad esempio, EDITA QUADRATO, e apportare qualche modifica alla procedura: non c'è da aver paura di rovinarla, perché poi si può rimettere tutto a posto come prima. Ecco alcune proposte:

- Modificare una delle istruzioni di QUADRATO. Terminare la fase di composizione con la battuta della coppia di tasti CTRL-C e provare a richiamare il comando QUADRATO, così modificato, per vedere che cosa succede. Ripristinare poi la procedura QUADRATO come era all'inizio.
- Sostituire il numero assegnato ai comandi AVANTI, che è 50, con un altro valore e vedere che effetto si ottiene richiamando poi il comando QUADRATO.
- Sostituire gli angoli di rotazione nei comandi DESTRA, che valgono 90, con un nuovo valore. Richiamare ancora il comando QUADRATO e osservarne l'effetto.
- Cambiare il nome della procedura. Richiamare il nuovo nome e provare a richiamare ancora il vecchio. Battere poi il comando FV TITOLI, per vedere l'elenco delle procedure assegnate al computer.
- Riprendere la procedura QUADRATO, riportandola allo stato iniziale.

SUGGERIMENTO

L'impiego dell'editor (così si chiama anche lo strumento di composizione di testi del Logo) può sembrare all'inizio complicato. La cosa più importante da capire è il comportamento del cursore: qualunque carattere battuto alla tastiera viene inserito nel testo là dove si trova il cursore, e tutto il resto viene spostato in modo da fare spazio. Se, al contrario, si cancella una parte di testo, il cursore si sposta all'indietro, portando con sé il testo che rimane alla sua destra.
Le due operazioni di composizione sono:

1. Lo spostamento del cursore senza modificare il testo.
2. La modifica del testo, che può consistere nella cancellazione di una parte, nella battitura di nuovi caratteri, o nella creazione di una nuova riga (la modifica del testo ha anche l'effetto di spostare il cursore).

All'inizio, molti commettono errori di composizione dovuti proprio alla confusione fra questi due tipi di operazione, o alla comprensione errata di come funziona il cursore. Ecco qui alcuni problemi tipici di chi inizia:

- Usare la barra di spaziatura, invece del tasto →, per spostare il cursore verso destra, o cercare di inserire uno spazio battendo il tasto →.
- Usare il tasto RETURN invece del tasto ↓ per spostarsi di una riga verso il basso. Molti hanno infatti l'abitudine di battere il tasto RETURN alla fine di ogni riga: facendo ciò con l'editor, si aggiunge una nuova riga, spostando in basso le altre. Se questo succede per sbaglio, si può subito rimediare battendo il tasto INST/DEL.
- Ignorare dove va posto il cursore per inserire o per cancellare una parte di testo.

Chi già conosce il linguaggio BASIC può incontrare un'ulteriore difficoltà: in BASIC infatti se si batte un testo su una riga già scritta, si sostituisce il vecchio contenuto con il nuovo, mentre invece il Logo inserisce il nuovo testo facendo scorrere il vecchio in avanti.
Succede anche di dimenticare la battuta conclusiva della coppia di tasti CTRL-C, o del tasto RUN/STOP, per ripristinare il modo operativo normale.
Il Logo ha un repertorio di comandi di composizione molto ricco. Bisognerebbe però incominciare a far pratica con pochi comandi, e completare poi la conoscenza degli altri con gradualità. In effetti, per comporre testi occorrono solo pochi tasti: i tasti di spostamento (segnati con le freccette), il tasto INST/DEL e il tasto RETURN (se si batte un tasto di spostamento alla fine di una riga, si sposta il cursore verso l'alto, o verso il basso).
Tutti gli altri tasti di composizione offrono solo un modo più rapido e co-

modo per fare le stesse cose: può essere utile preparare un elenco di tali tasti, da tenere bene in vista, in modo che possano essere provati da chi lo desidera, senza difficoltà.
Ancora un cenno sui termini tecnici in gergo (carattere, testo, cursore, editing): essi non sono superflui, ma servono a definire meglio i concetti e ad afferrarne le differenze nell'impiego pratico, anche se a prima vista possono sembrare strani.

ALTRE COMBINAZIONI DI TASTI PER LA COMPOSIZIONE

La composizione viene resa più svelta e comoda se si ricorre ad alcuni altri tasti, che comunque vanno usati solo quando si padroneggiano i tasti di base, visti prima. All'inizio, conviene forse saltare questa parte, e passare direttamente al paragrafo 4.3, che riguarda la registrazione su disco delle procedure. Più tardi, quando si avrà maggiore confidenza con la tecnica di editing, si potrà riprendere l'argomento con maggior profitto. Vediamo quali sono questi tasti e la loro funzione:

CTRL-L
Sposta il cursore alla fine della riga.

CTRL-A
Sposta il cursore all'inizio della riga.

CTRL-F
Sposta il cursore in avanti per un intero schermo, se questo è completamente occupato da testo.

CTRL-B
Sposta il cursore all'indietro di un intero schermo, se questo è completamente occupato da testo.

CTRL-D
Cancella il carattere su cui si trova il cursore, senza fare arretrare quest'ultimo (D è l'iniziale di Distrugge).

CTRL-K
Cancella la riga su cui si trova il cursore.

CTRL-O
Crea una nuova riga, all'altezza del cursore, spostando in basso il testo che resta.

ALTRI ASPETTI DELL'EDITOR

Si può usare l'editor per assegnare al computer diverse procedure in una volta sola. Basta proseguire con la procedura successiva, dopo aver battuto l'istruzione FINE (o, in alternativa, END) che conclude la procedura in corso di composizione. Alla fine di tutto, si batte CTRL-C.
Se si batte EDITA e subito dopo anche il tasto RETURN, senza indicare il nome di una procedura, il computer prende ciò che al momento compare sullo schermo e si predispone alle normali operazioni di composizione. Se si batte EDITA in questo modo subito dopo aver usato la tartaruga, si inizia la fase di composizione con lo schermo vuoto. Si può anche fare la composizione di più procedure in una volta sola, battendo il comando EDITA seguito dall'elenco dei nomi delle procedure che interessano, racchiuso fra parentesi quadre, come si vede qui di seguito:

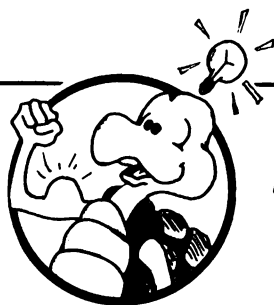
EDITA [QUADRATO, FINESTRA] (e RETURN)

Per comporre tutte le procedure contemporaneamente, si può battere il comando:

EDITA TUTTO (e RETURN)

Per avere maggiori informazioni sull'impiego dell'editor, conviene leggere il Manuale del Logo Commodore 64.

4.3 Come registrare le procedure su un disco di servizio

Il computer dispone di due tipi di memoria: la memoria di lavoro, con la quale vengono eseguite tutte le elaborazioni e che però non conserva nulla quando il computer viene spento, e la memoria permanente, registrata su disco, che quando occorre va letta (o, come anche si dice, caricata) e trasferita nella memoria di lavoro.

La memoria del computer non è molto diversa dalla memoria di una persona. Anche noi ricordiamo certe cose all'istante (ecco la nostra memoria di lavoro) e, per altre cose, dobbiamo consultare appunti, libri, o altri documenti (ecco la nostra memoria permanente).
Ad esempio, riusciamo a ricordare solo alcuni numeri di telefono, e quando ne impariamo uno nuovo, in genere lo dobbiamo annotare sulla nostra agenda: per di più, esiste anche l'elenco telefonico, che contiene migliaia e migliaia di numeri telefonici, che può essere consultato in qualunque momento (però i numeri rimangono sempre quelli, e non possono essere cambiati da chi li legge).
Ecco, i numeri di telefono che si ricordano a mente appartengono alla nostra memoria di lavoro, e sono come le procedure appena assegnate al Logo. I numeri di telefono annotati sull'agenda personale sono invece come le procedure assegnate al Logo e registrate sul disco di servizio. L'elenco del telefono è invece simile al disco che contiene il linguaggio Logo, che può essere sempre letto ma mai modificato.

Dopo aver assegnato al computer qualche nuova procedura, bisogna registrarla sul disco di servizio: se non si ha già un disco di servizio per il Logo, bisogna crearne uno seguendo le indicazioni dell'Appendice B.
Il Logo registra le informazioni sul disco di servizio disponendole in archivi elementari, detti file. Ecco i principali comandi dedicati a questa attività:

CONSERVA "LAURA
Registra tutte le procedure che si trovano nella memoria di lavoro del computer, e le colloca in un file chiamato "LAURA del disco di servizio.

RECUPERA "LAURA
Legge dal disco di servizio tutte le procedure registrate nel file chiamato "LAURA e le trasferisce nella memoria di lavoro del computer.

CATALOGO
Scrive l'elenco di tutti i file registrati sul disco di servizio.

CANCEARC "LAURA
Cancella dal disco di servizio il file chiamato "LAURA. Naturalmente, bisogna fare molta attenzione prima di cancellare un file, per essere ben sicuri di non averne più bisogno.
In pratica, per registrare un nuovo file si incomincia con la scelta di un nome nuovo: tanto per incominciare, si può scegliere il proprio nome. In uno stesso file si possono registrare anche molte procedure diverse.
Si prende dunque il disco di servizio e lo si inserisce nel drive, chiudendo poi lo sportello e battendo:

FV TITOLI (e RETURN)

Si ottiene in questo modo l'elenco delle procedure presenti nella memoria di lavoro, poi si batte:

CONSERVA "LAURA (e RETURN)

Così le procedure presenti in memoria sono registrate nel file chiamato "LAURA: si accende la luce rossa del drive, che si mette a ronzare per un po'. Alla fine compare sullo schermo il segno ? con cui il computer segnala di essere pronto a nuovi comandi.

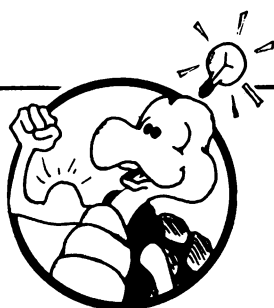
TRABOCCHETTO

Bisogna fare attenzione a battere le virgolette, ", appena prima del nome scelto, senza lasciare spazi vuoti. Alla fine le virgolette non vanno mai chiuse.
In pratica, bisogna battere, ad esempio, "LAURA e non " LAURA o "LAURA".
Se qualcosa non funziona nel verso giusto, conviene controllare se il disco è inserito bene nel drive, se lo sportello è chiuso, e se il disco di servizio è stato predisposto correttamente. Se i problemi persistono, conviene chiedere aiuto a chi ne sa di più.

Si può ora provare un nuovo comando del Logo, battendo:

CATALOGO

Il computer presenta allora l'elenco di tutti i file registrati sul disco di servizio. Tutti i file che contengono procedure del Logo hanno nomi che terminano con .LOGO:

```
?CATALOGO
B 003 CERCHIETTO.LOGO
B 003 CERCHI.LOGO
B 008 GARA.LOGO
B 002 DISTANZA.LOGO
B 002 LEGGITASTO.LOGO
B 003 LEGGINUMERO.LOGO
B 002 ASPETTA.LOGO
B 003 RETTANGOLI.LOGO
B 003 ACASO.LOGO
B 012 POETA.LOGO
B 006 SCHIZZO.LOGO
B 007 MATQUIZ.LOGO
B 003 TROVANUMERO.LOGO
B 009 TIRO.A.SEGNO.LOGO
?
```

IDEA

Ecco un paragone utile: un file registrato su disco è come un raccoglitore in una libreria. Il raccoglitore contiene molti fogli, su ciascuno dei quali è scritta una procedura. I cassetti estrabili della libreria possono contenere più raccoglitori, ciascuno però segnato con un nome diverso, in modo che sia facile trovarlo, quando lo si cerca. Inoltre, molte persone possono registrare i propri file sullo stesso disco, proprio come se fossero raccoglitori conservati nello stesso cassetto.

Se si vede comparire nell'elenco del comando CATALOGO il nuovo nome, appena assegnato a un file, si è certi che le procedure in memoria sono state registrate sul disco. D'ora in poi, quando se ne ha bisogno, basta leggerle dal disco e trasferirle in memoria.
A questo punto, si possono provare i seguenti comandi:

FV TITOLI
Fornisce l'elenco delle procedure appena registrate.

CANCELLA TUTTO
Ripulisce la memoria di lavoro del computer.

FV TITOLI
A questo punto, dopo aver ripulito la memoria del computer, non si trova più nessuna procedura!

RECUPERA "LAURA
Legge da disco il file "LAURA (o un altro, se si è usato un nome diverso quando lo si è registrato).

FV TITOLI
Dopo la lettura, si trovano in memoria le procedure trasferite dal disco come si vede dall'elenco prodotto con il comando FV TITOLI.

Di solito, quando si riprende il Logo si legge per prima cosa il disco di servizio, trasferendo in memoria le procedure lì registrate, nel file che interessa (il primo file si può chiamare con il proprio nome, ad esempio). Ecco come si può procedere:

1. Per prima cosa, si batte CIAO oppure CANCELLA TUTTO, per ripulire la memoria di lavoro del computer.
2. Si legge poi il file che interessa, battendo RECUPERA "LAURA (se il file si chiama "LAURA).
3. Se si è assegnata al computer una nuova procedura, la si registra su disco (dopo averla provata per bene) con tutte le altre procedure presenti in memoria, battendo il comando CONSERVA "LAURA (se il file si chiama "LAURA).

Quando si smette di usare il Logo, conviene battere CIAO: così si libera la memoria di lavoro del computer per chi lo userà dopo di noi.

TRABOCCHETTO

ATTENZIONE! Se non si legge il proprio file dal disco di lavoro quando si inizia ad usare il computer, succede un pasticcio: quando infatti si registrano le nuove procedure, si cancellano tutte quelle già registrate. Per prudenza, bisogna allora leggere sempre da disco tutte le procedure, perché così, quando si registra di nuovo il file di servizio, queste vengono ricopiate insieme a quelle nuove, o a quelle modificate.

SUGGERIMENTO

La registrazione su disco delle procedure può essere per alcuni piuttosto difficile da capire. Spesso torna utile distinguere due tipi di memoria: la memoria a breve termine, o memoria di lavoro, che contiene tutto ciò che il Logo conosce nel momento attuale, e la memoria permanente, contenente informazioni che il Logo nel momento attuale non conosce, ma che può recuperare nel modo descritto in questo capitolo.
I paragoni dell'elenco del telefono e della libreria sono molto utili per presentare a chi impara il Logo il concetto di registrazione. C'è però una differenza da tenere ben presente, fra il file su disco e l'agenda personale dei numeri telefonici: quando si registrano le procedure su disco, il computer trasferisce tutto il contenuto della memoria. Inoltre, la registrazione su disco non altera il contenuto della memoria di lavoro.
Viceversa, quando si legge un file, si trasferisce l'intero suo contenuto nella memoria di lavoro del computer. Il file su disco rimane inalterato, fino a che non si fa una nuova registrazione con lo stesso nome di file: in quel momento, viene cancellato il vecchio file e ne viene creato uno nuovo. Ecco perché è così importante leggere il vecchio file prima di incominciare a lavorare con il Logo: così infatti la memoria di lavoro contiene tutte le informazioni

del file, che vengono quindi trasferite su disco con la registrazione successiva, insieme alle nuove procedure e alle modifiche eventualmente fatte.
Succede, qualche volta, nonostante tutte le precauzioni, che qualcosa vada perso. Conviene sempre tenere una copia su carta del contenuto del disco, proprio per poter recuperare le informazioni perse per sbaglio. Nel paragrafo 4.5 si possono trovare le indicazioni per stampare il contenuto del disco, con una stampante. Un'altra buona idea è fare di tanto in tanto una copia del disco di servizio (seguendo le indicazioni dell'Appendice B).

LAVORARE CON PIÙ FILE

Questa parte torna utile quando la memoria di lavoro incomincia a riempirsi di procedure, o quando vanno creati file di servizio diversi, per diversi progetti.
In effetti, si può lavorare con diversi file, proprio come si usano diversi raccoglitori, intestati ciascuno con un proprio nome, per tenere le carte di argomenti diversi. L'impiego di più file di servizio diventa utile quando si sono ormai definite molte procedure: conviene allora raggrupparle per argomento (ad esempio un particolare disegno, come nel Capitolo 6), oppure per tipo di funzione (che può essere la realizzazione di un certo tipo di figura, come nel Capitolo 5).
Quando si lavora con più file, bisogna fare il possibile per tenerli ben separati, in modo da non fare confusione. All'inizio di un nuovo progetto, conviene stabilire il nome da usare per il file di servizio, da usare poi sempre per registrare le procedure. Quando poi si riprende il lavoro su quel progetto, bisogna sempre per prima cosa leggere il file di servizio. Se si passa ad un altro progetto, si deve inoltre ripulire la memoria di lavoro con il comando CANCELLA TUTTO, oppure con CIAO, e leggere il file di servizio che interessa.

4.4 La registrazione di figure

Le figure realizzate con il Logo possono essere registrate su file, direttamente, e da qui lette e trasferite di nuovo in memoria. Naturalmente, ogni figura va individuata con un proprio nome. Ecco i comandi da usare:

CONSERDIS "QUADRATI
Registra su disco una figura, con il nome "QUADRATI.

RECUPDIS "QUADRATI
Legge dal disco la figura chiamata "QUADRATI, e la disegna sullo schermo.

CANCEDIS "QUADRATI
Cancella dal disco la figura chiamata "QUADRATI.

Se si batte il comando CATALOGO dopo aver registrato una figura grafica, se ne vede comparire il nome per due volte, nell'elenco: se il nome è ad esempio QUADRATI, si vede QUADRATI.DIS1 e QUADRATI.DIS2. Il termine .DIS significa che il file contiene un disegno.
La registrazione di figure grafiche occupa molto spazio sul disco, perciò ogni tanto bisogna cancellare le figure vecchie, se non servono più.

4.5 La stampante

Se si ha una stampante VIC-1525E collegata al computer Commodore 64 alla porta seriale 4, si è in grado di stampare su carta una copia delle procedure, e persino di ottenere copia delle figure disegnate con la tartaruga del Logo.
Per incominciare, bisogna accendere la stampante. Ecco poi i comandi a disposizione:

STAMPANTE
Serve per predisporre la stampante, sostituendola allo schermo. Dopo questo comando, i normali comandi di scrittura del Logo agiscono sulla stampante, invece che sullo schermo.

STAMPA [CIAO]
Stampa CIAO sulla stampante.

FAIVEDERE TITOLI
Stampa l'elenco delle procedure in memoria.

FV QUADRATO
Stampa le istruzioni che formano la procedura QUADRATO.

FV TUTTO
Stampa le istruzioni di tutte le procedure in memoria.

CATALOGO
Stampa l'elenco di tutti i file registrati su disco.

NOSTAMPANTE
Serve per disabilitare la stampante, ripristinando la scrittura sullo schermo.

LA STAMPA DELLE FIGURE

Con la stampante VIC-1525E si possono stampare anche le figure disegnate con il Logo, usando il file STAMPADIS del disco di utilità del Logo. Conviene seguire le indicazioni del manuale Logo del Commodore 64.

SUGGERIMENTO

La stampa delle procedure e delle figure realizzate con il Logo è senz'altro un momento importante nell'apprendimento del Logo. La disponibilità di una copia scritta su carta delle procedure, da inserire nel diario o da affiggere su un tabellone accanto al computer, può essere un bell'aiuto per chi insegna il Logo ad un gruppo di allievi, anche perché questi vengono facilitati molto nel loro apprendimento.
Non è necessario, in una scuola, avere una stampante per ogni computer: in molti casi, ne basta una sola, installata magari in biblioteca, dove i ragazzi possono andare per stampare i loro file.

5

Costruzioni grafiche con la tartaruga

In questo capitolo si vedono numerosi esempi di figure grafiche realizzate con la tartaruga, che possono essere comodamente copiate o anche modificate. Si vede anche come creare e impiegare delle procedure autocostruite.
Questo capitolo può essere scorso alla svelta, per una prima rassegna delle idee che propone, oppure anche letto con attenzione e approfondito, come fonte di spunti per molte possibili costruzioni grafiche con la tartaruga. Occorre avere un certo spirito di iniziativa, e non lasciarsi scoraggiare, se una costruzione si rivela più difficile di quanto sembri a prima vista; conviene inoltre avere fiducia in se stessi, e affrontare anche gli esempi più impegnativi. Comunque, molte costruzioni sono più semplici di quanto sembrano, se le si prende per il loro verso.

Comando	Abbreviazione	Esempio
RIPETI		RIPETI 4 [AVANTI 20 DESTRA 90]

Viene usato il file "CERCHI (vedi Appendice A).

5.1 Procedure principali e secondarie

Una procedura è un comando nuovo assegnato da noi al computer, che può essere richiamato come un qualunque altro comando originale del Logo. È anche possibile inserire, all'interno di una procedura, una proce-

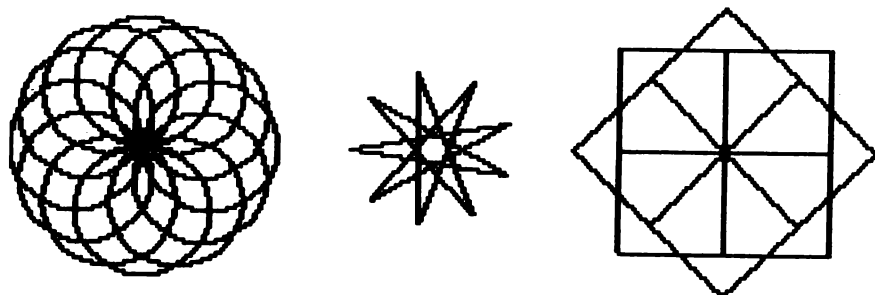



Figura 5.1 Figure tracciate con la tartaruga

dura secondaria (e all'interno di questa, un'altra ancora, e così via). Un esempio dimostrativo può essere quello di Figura 5.2, che viene costruito con molta facilità se già si dispone di una procedura per tracciare quadrati.

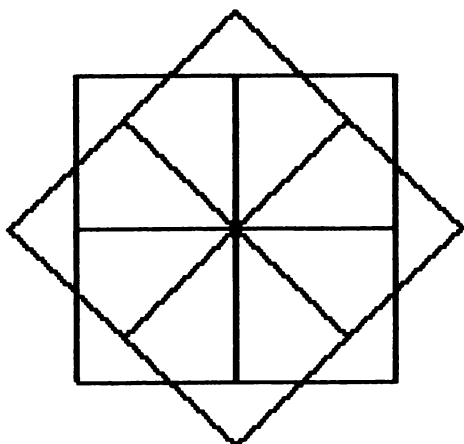



Figura 5.2 Stella costruita con due griglie ruotate

Se ancora non si ha una procedura per tracciare quadrati, la si può assegnare al Logo con queste istruzioni:

```
PER QUADRATO
AVANTI 50
DESTRA 90
AVANTI 50
DESTRA 90
AVANTI 50
```

```
DESTRA  90
AVANTI  50
DESTRA  90
FINE
```

Figura 5.3 Quadrato tracciato con la procedura QUADRATO

La procedura QUADRATO ha una struttura molto semplice, che è basata sulla ripetizione della coppia di istruzioni AVANTI 50, DESTRA 90, fino a che la tartaruga non ritorna al punto di partenza. Con questa procedura si può poi costruire la procedura GRIGLIA, battendo le seguenti istruzioni:

```
PER GRIGLIA
QUADRATO
SINISTRA 90
QUADRATO
SINISTRA 90
QUADRATO
SINISTRA 90
QUADRATO
SINISTRA 90
FINE
```

Figura 5.4 Effetto della procedura GRIGLIA

La GRIGLIA, come si vede, risulta dalla ripetizione della coppia di istruzioni QUADRATO, SINISTRA 90, fino a che la tartaruga ritorna al punto di partenza. Ora che si ha la GRIGLIA, si può costruire la stella di Figura 5.5, in questo modo:

```
TO STELLA
GRIGLIA
DESTRA 45
GRIGLIA
FINE
```

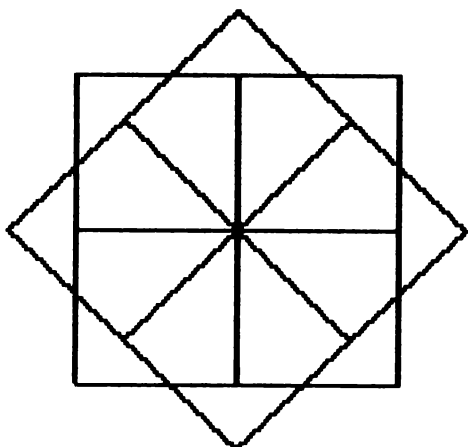
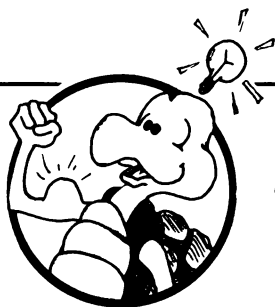
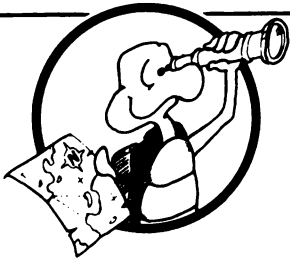



Figura 5.5 Effetto della procedura STELLA, che ricorre alla procedura GRIGLIA

L'impiego delle procedure secondarie, all'interno di altre procedure, è una delle tecniche più potenti che si possono imparare con il Logo. Senza le procedure QUADRATO e GRIGLIA, il disegno di Figura 5.2 avrebbe richiesto molto tempo e molta attenzione. L'uso delle procedure secondarie, in generale, facilita la realizzazione dei progetti, e la loro compren-

sione, specie dopo che è trascorso un certo tempo, e per chi si trova a leggere le istruzioni scritte da un altro.

ESPLORAZIONE

Sono molti i disegni che sembrano a prima vista complicati, e che invece si possono costruire abbastanza facilmente con l'impiego di procedure secondarie. Come esempio, si possono provare i disegni di Figura 5.6.

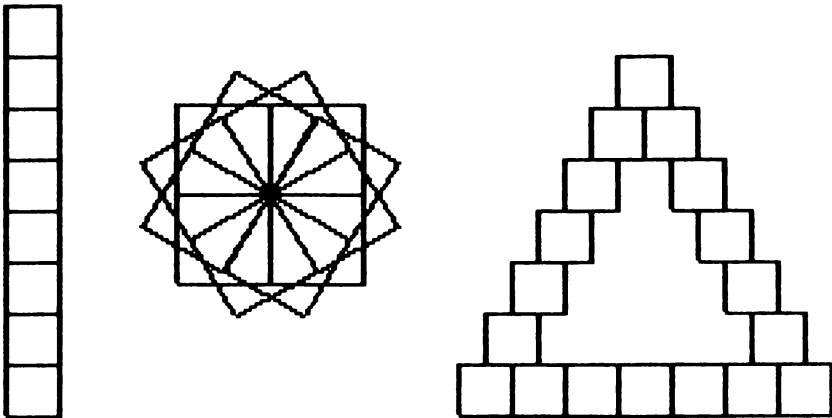



Figura 5.6 Disegni costruiti con quadrati

SUGGERIMENTO

Molti degli esempi proposti diventano ancora più facili da costruire con il comando RIPETI, e con le tecniche recursive descritte nei paragrafi 5.3 e 5.4. Qui gli esempi servono però per dimostrare l'utilità delle procedure secondarie, ed è per questo che vengono proposti senza ricorrere a tecniche recursive. Se chi sta imparando cerca un modo più facile di realizzare gli esempi, si può senz'altro indirizzare ai paragrafi appena citati.
Il concetto di procedura secondaria è fondamentale per il Logo, ed è in parte per suo merito che il Logo è molto più facile di tanti altri linguaggi di programmazione. In questo modo infatti, chi programma si può costruire un proprio linguaggio, usando i comandi così definiti per eseguire compiti altrimenti molto lunghi e difficili. Ogni nuova procedura è indipendente dalle altre, e può venire inserita in più di un progetto applicativo.
Inoltre, una procedura può anche comparire in se stessa, come procedura secondaria: questo può sembrare un vero assurdo, ma invece rappresenta uno strumento veramente unico e di estrema potenza, come si vedrà nel paragrafo 5.4, che tratta delle tecniche recursive.

5.2 Figure regolari

Una figura si dice regolare quando ha i lati e gli angoli uguali fra loro: sono figure regolari il quadrato e il triangolo equilatero, ad esempio. Per tracciare un quadrato con la tartaruga, basta ripetere la coppia di istruzioni AVANTI (con l'avanzamento che si vuole) e DESTRA 90 finché la tartaruga non raggiunge di nuovo il punto di partenza. Si può tracciare una grande varietà di figure di questo tipo, con un'opportuna scelta degli angoli di rotazione. Ad esempio, si può disegnare una stella così:

```
AVANTI 50
DESTRA 150
AVANTI 50
DESTRA 150
..........
```

Si prosegue con la coppia di istruzioni AVANTI 50 e DESTRA 150 finché la stella non è finita. Alla fine, si può registrare il disegno su disco, con il nome che si preferisce.

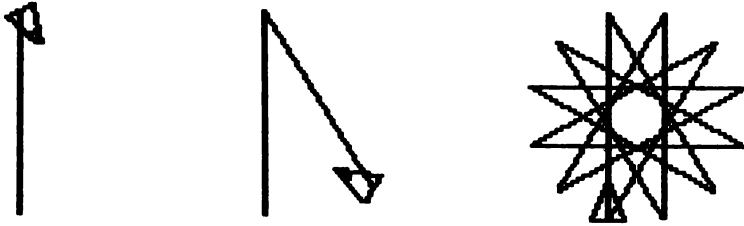
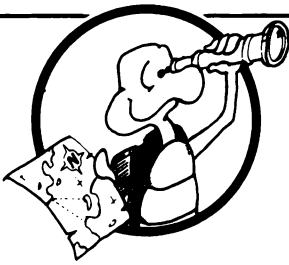



Figura 5.7 Passi di costruzione di una stella

ESPLORAZIONE

Si può realizzare l'immagine speculare della stella di Figura 5.7, facendo ruotare la tartaruga verso sinistra (con l'istruzione SINISTRA 150). Altre forme ancora si ottengono assegnando altri angoli di rotazione.
Con questa tecnica si può anche tracciare un triangolo equilatero. La parte più difficile riguarda la scelta del giusto angolo di rotazione: conviene provare da soli, cercando magari di applicare ciò che già si conosce di geometria. Comunque, l'angolo di rotazione di 90 gradi è insufficiente, come si vede battendo:

```
AVANTI 50 DESTRA 90
AVANTI 50 DESTRA 90
AVANTI 50 DESTRA 90
```

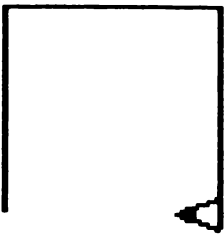



Figura 5.8 Tre angoli di 90 gradi non bastano per tracciare un triangolo

Invece, l'angolo di rotazione di 150 gradi è eccessivo, come si vede subito battendo:

```
AVANTI 50 DESTRA 150
AVANTI 50 DESTRA 150
AVANTI 50 DESTRA 150
```

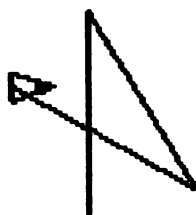



Figura 5.9 Tre angoli di 150 gradi sono troppi per un triangolo

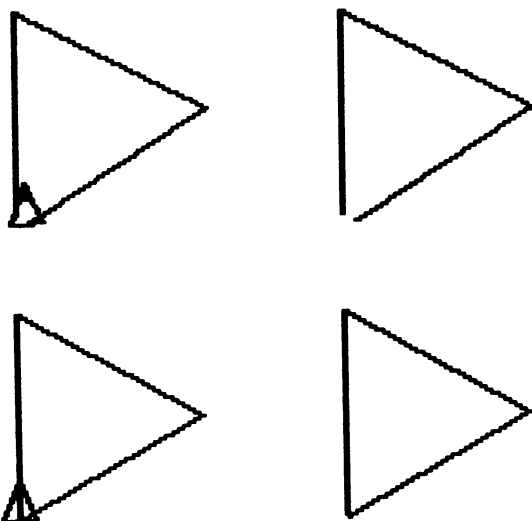



Figura 5.10 Escludendo la visualizzazione della tartaruga, si può apprezzare meglio la coincidenza dei punti di partenza e di arrivo

L'angolo giusto è compreso fra 90 e 150 gradi: si può continuare a provare, finché si trova l'angolo che fa tornare la tartaruga al punto di partenza esattamente con tre passi.

TRABOCCHETTO

Se si prova con un angolo molto prossimo a quello giusto, si può avere difficoltà a distinguere se la tartaruga ritorna proprio al punto di partenza. Comunque, la coincidenza si nota meglio se si esclude la visualizzazione della tartaruga.

5.3 Il comando RIPETI

Il comando RIPETI fa ripetere al computer un elenco di comandi, quante volte si vuole. Questo comando è molto utile quando si sa già quante volte va ripetuta una certa sequenza di comandi. Il comando RIPETI richiede due indicazioni: la prima è il numero di ripetizioni da eseguire, la seconda è la sequenza di comandi da ripetere, che va racchiusa fra parentesi quadre, come si vede negli esempi indicati qui di seguito:

```
RIPETI  4[AVANTI 20 SINISTRA 90]
RIPETI 12[QUADRATO DESTRA 30]
```

Il comando RIPETI può essere usato nelle procedure QUADRATO e GRIGLIA del paragrafo 5.1:

```
PER QUADRATO1
RIPETI 4[AVANTI 50 DESTRA 90]
FINE

PER GRIGLIA1
RIPETI 4[QUADRATO1 SINISTRA 90]
FINE

PER STELLA1
RIPETI 2[GRIGLIA1 DESTRA 45]
FINE
```

Il comando RIPETI torna molto utile anche per tracciare figure regolari (quadrati, triangoli equilateri, ecc.); ecco qui alcuni esempi:

```
PER STELLA150
RIPETI 12[AVANTI 100 DESTRA 150]
FINE

PER TRIANGOLO
RIPETI 3[AVANTI 80 DESTRA 120]
FINE
```

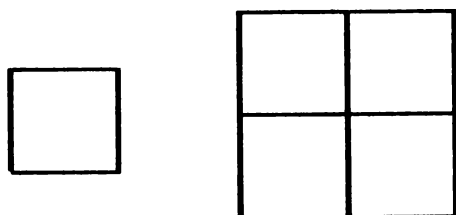



Figura 5.11 Effetto dell'impiego del comando RIPETI

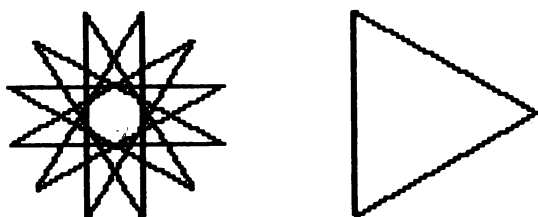



Figura 5.12 Stella e triangolo costruiti con l'impiego del comando RIPETI

Con RIPETI, si può anche trovare l'angolo da definire per ottenere la figura che interessa. Ad esempio, quale angolo va scelto per ottenere la stella a cinque punte di Figura 5.13?

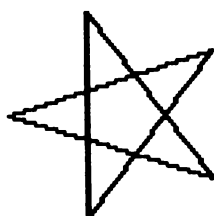



Figura 5.13 Stella a cinque punte

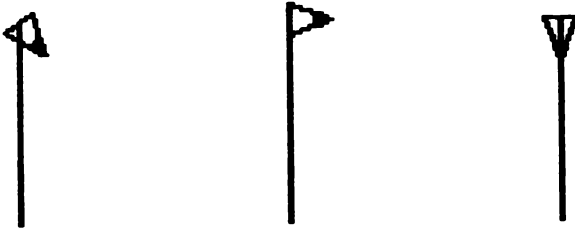



Figura 5.14 L'angolo corretto è compreso fra 90 e 180 gradi

Si può ricorrere al comando:

RIPETI 5[AVANTI 50 DESTRA *angolo*]

In un primo tempo, si assegna all'angolo un valore di tentativo, e si vede l'effetto ottenuto (comunque, l'angolo cercato è maggiore di 90 gradi e minore di 180 gradi, come si scopre subito riflettendo sulla Figura 5.14). Si può incominciare con 120 gradi, aumentandolo poi a poco a poco, per approssimare la stella:

RIPETI 5[AVANTI 80 SINISTRA 120]

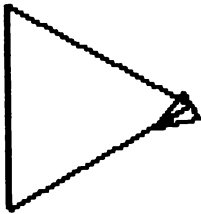



Figura 5.15 L'angolo di 120 gradi è troppo piccolo

RIPETI 5[AVANTI 80 SINISTRA 130]

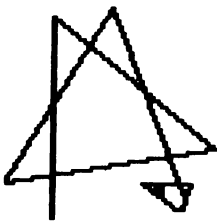



Figura 5.16 Anche 130 gradi sono insufficienti

RIPETI 5[AVANTI 80 SINISTRA 140]

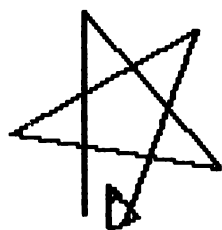



Figura 5.17 L'angolo di 140 gradi va meglio

Come si vede, l'angolo di 140 gradi va abbastanza bene. Si può allora provare con 150 gradi:

RIPETI 5[AVANTI 80 SINISTRA 150]

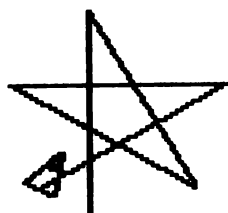



Figura 5.18 L'angolo di 150 gradi è eccessivo

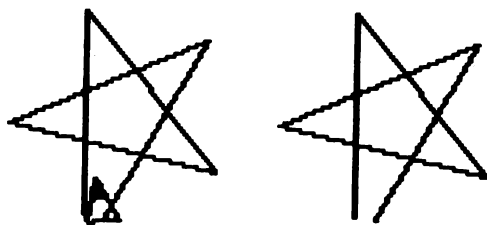
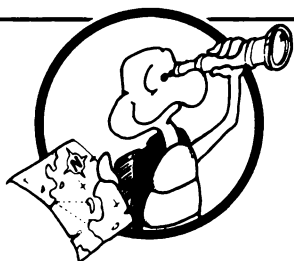



Figura 5.19 La tartaruga può nascondere il punto di incontro: a volte bisogna escluderla

Con l'angolo di 150 gradi, l'ultimo lato incrocia il primo: allora bisogna tornare un poco indietro. L'angolo giusto è compreso fra 140 e 150 gradi: quanto vale esattamente? Si può provare da soli.

TRABOCCHETTO

Se si ha difficoltà a distinguere il punto di incontro, si può escludere la visualizzazione della tartaruga.

ESPLORAZIONE

Ecco ancora altre possibili applicazioni del comando RIPETI:

- Usare RIPETI con procedure secondarie che costruiscono quadrati, stelle e triangoli.

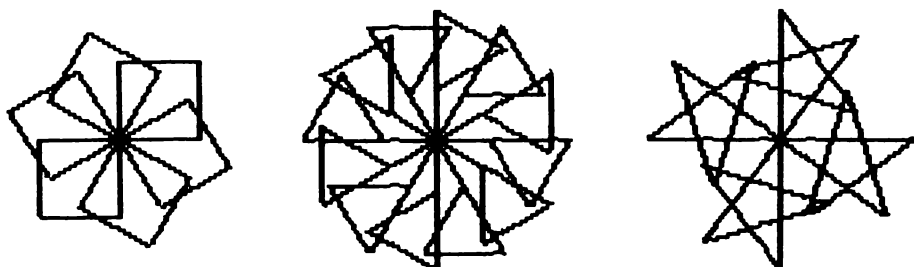



Figura 5.20 Figure ottenute con il comando RIPETI

- Costruire figure speculari. Ad esempio, un quadrato con la tartaruga rivolta a destra e un secondo con la tartaruga rivolta a sinistra. O anche triangoli e stelle, fra loro speculari.
- Costruire altre figure regolari, come l'esagono, o la stella a otto punte.

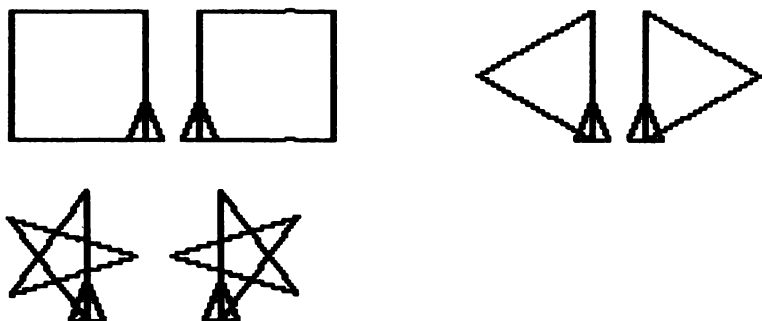



Figura 5.21 Figure speculari

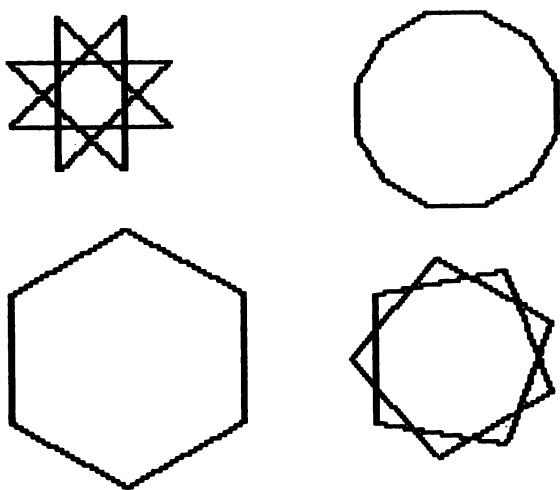



Figura 5.22 Poligoni regolari e stelle costruiti con il comando RIPETI

SUGGERIMENTO

La stella a cinque punte presentata prima è un buon esempio di soluzione sistematica di problemi di costruzione grafica. Se ne possono trarre tre buone idee:

1. Semplificare il problema. Vi sono talmente tante possibili forme di stella a cinque punte che sarebbe molto difficile disegnarne una per tentativi. Scegliendo invece una forma regolare (cioè con gli angoli e i lati uguali fra loro) si rende il problema più abbordabile. In questo modo inoltre si riesce spesso a trarre vantaggio da altri problemi già risolti, come ad esempio la costruzione del quadrato o del triangolo equilatero.
2. Limitare i tentativi ad un solo elemento per volta: in questo caso, l'angolo della stella a cinque punte. Un difetto molto diffuso, comune a grandi e piccoli, è la cattiva delimitazione dei tentativi. Conviene osservare qualcuno che tenta di risolvere questo problema: è facile che provi a cambiare molti elementi in una volta sola (ad esempio, l'angolo e la grandezza della stella, oltre all'orientamento), con il risultato di fare una gran confusione e di non venire a capo di nulla.
3. Provare con una stima iniziale, cercando man mano di restringere il campo da esplorare. Ad esempio, sapere che l'angolo giusto è compreso fra 140 e 150 gradi è molto meglio che saperlo compreso fra 90 e 180 gradi. Inoltre, se si fanno tentativi a caso, non in modo sistematico, si può perdere molto tempo prima di trovare la soluzione.

Alcuni hanno più occhio di altri, ma non è detto che riescano a risolvere un problema più in fretta di chi, magari, ha meno istinto ma un metodo di ricerca più sistematico. Nell'esempio della stella a cinque punte si vede come è possibile avvicinarsi abbastanza in fretta alla soluzione, se si segue un buon metodo di indagine.
Il problema della stella a cinque punte può essere affrontato in un modo del tutto diverso, con un approccio analitico. Bisogna sapere (oppure scoprire) che la tartaruga, per tracciare una qualunque figura chiusa, deve necessariamente ruotare di un angolo di 360 gradi, o di un suo multiplo intero. Questa proprietà, che per la tartaruga è sempre vera, può essere sfruttata con gran vantaggio. Ad esempio, dato che la tartaruga esegue cinque rotazioni per tracciare la stella, si può provare con l'angolo ampio 360/5, per proseguire con l'angolo di (2*360)/5, e così via fino a che non si traccia la stella. Il calcolo degli angoli può essere lasciato ai comandi aritmetici del Logo, come si vede qui:

```
RIPETI 5[AVANTI 80 SINISTRA 360/5]
RIPETI 5[AVANTI 80 SINISTRA (2*360)/5]
..........
```

5.4 Le tecniche recursive

C'è un modo molto pratico per far ripetere un'operazione al computer, più e più volte: scrivere una procedura che richiami, come procedura secondaria, una copia di se stessa. Questa tecnica recursiva è utile, ad esempio, quando si prova una figura nuova, senza sapere in anticipo per

quante volte va ripetuto un elemento grafico. Insomma, se si conosce l'angolo di rotazione ma si ignora il numero di ripetizioni, conviene senz'altro scrivere una procedura recursiva. Ecco qui un esempio:

```
PER STELLA160
AVANTI 100
DESTRA 160
STELLA160
FINE

PER FIORE
TRIANGOLO
DESTRA 60
FIORE
FINE
```

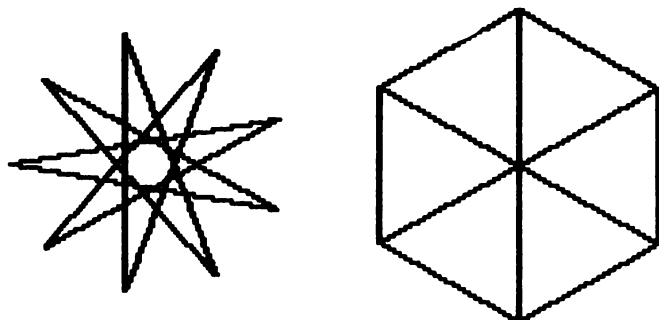



Figura 5.23 Stella e fiore tracciati con tecnica recursiva

Come funziona una procedura recursiva? Vediamola in dettaglio. Quando si batte il comando STELLA160, Logo chiama l'aiutante che porta quel nome. Subito, l'aiutante STELLA160 si mette al lavoro, seguendo le istruzioni assegnate. Richiama il comando AVANTI e fa spostare la tartaruga di 100 passi.
Poi, l'aiutante STELLA160 richiama il comando DESTRA, e gli fa ruotare la tartaruga di 160 gradi.
Infine, STELLA160 richiama un altro aiutante, che gli assomiglia come una goccia d'acqua, l'aiutante in seconda STELLA160, che ha anche lo stesso nome.
Questo aiutante in seconda STELLA160 è un vero e proprio replicante del primo, e si comporta esattamente come quello, perciò richiama AVANTI, poi DESTRA (che rispondono come al primo aiutante) e infine un nuovo STELLA160, come aiutante in terza, e così via.
Ma quando si finisce? Non si finisce mai. O meglio, per finire bisogna

battere un comando speciale alla tastiera, con la coppia di tasti CTRL-G. Conviene provare subito un esempio di procedura recursiva, come quella proposta qui. Più avanti, nel Capitolo 7, si vedrà come costruire una procedura recursiva che si arresta da sola. Per ora, bisogna accontentarsi di intervenire dall'esterno battendo CTRL-G.
Ecco ora qualcosa di divertente, che impiega la tecnica appena vista. Si traccia una figura qualunque, a caso (va bene anche una delle vecchie figure già tracciate) e la si rende recursiva, nel modo appena visto. Ad esempio:

```
PER PROVA
A 50
S 90
A 20
S 120
A 30
FINE
```

Figura 5.24 Ecco cosa traccia PROVA

Ecco come si rende PROVA recursiva:

```
PER PROVAUNO
PROVA
PROVAUNO
FINE
```

Si può anche fare qualcosa di più:

```
PER PROVADUE
PROVA
D 60
I 50
PROVADUE
FINE
```

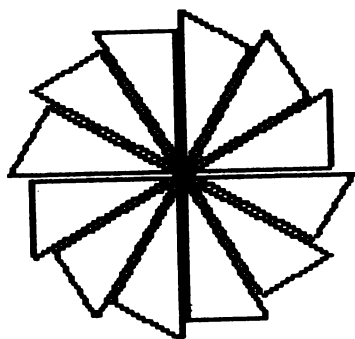



Figura 5.25 Ecco cosa traccia PROVAUNO

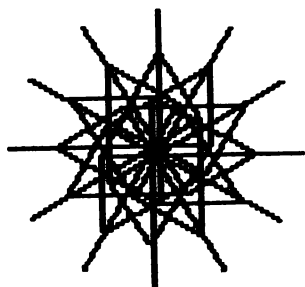
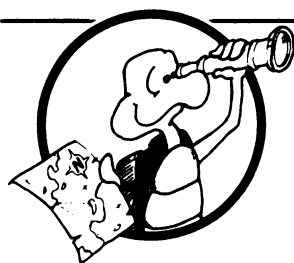



Figura 5.26 Ecco cosa traccia PROVADUE

Coraggio, ora conviene provare da soli. Basta un niente, e si può ottenere un disegno interessante.

Stelle e poligoni sono facili da costruire in modo recursivo. Le stelle sono figure regolari le cui linee si incrociano, come in STELLA135, i poligoni sono anch'essi figure regolari, ma i loro lati non si incrociano (come, ad esempio, in POLIGONO45).

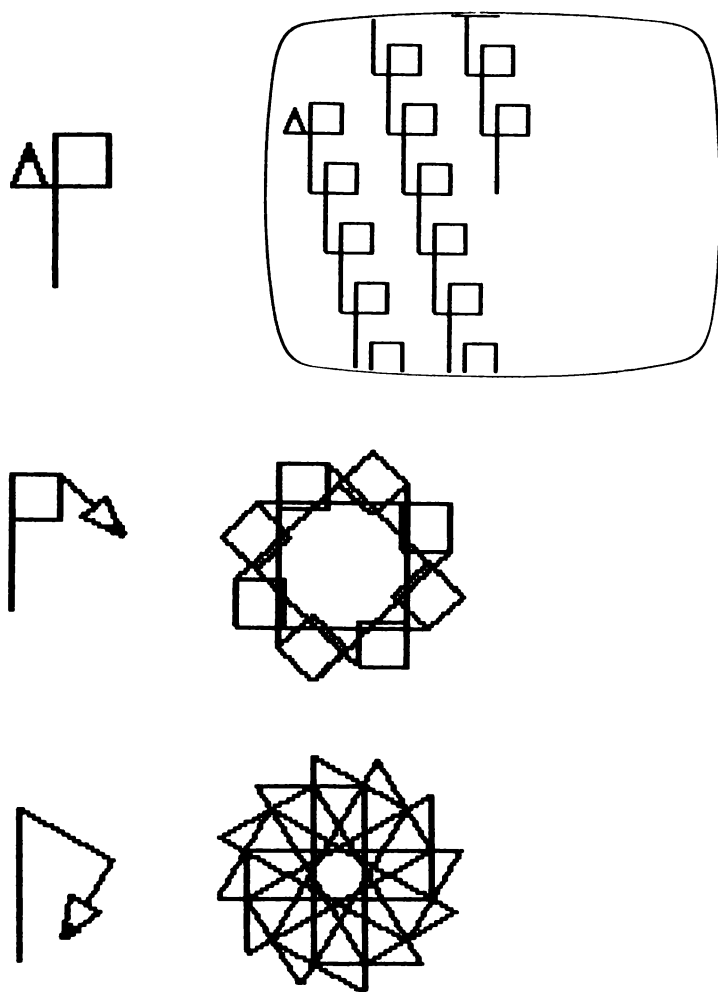



Figura 5.27 Motivi recursivi, con gli elementi grafici corrispondenti

Ecco le istruzioni di STELLA135:

```
PER STELLA135
A 60
S 135
STELLA135
FINE
```

Ecco invece le istruzioni di POLIGONO45:

```
PER POLIGONO45
A 30
D45
POLIGONO45
FINE
```

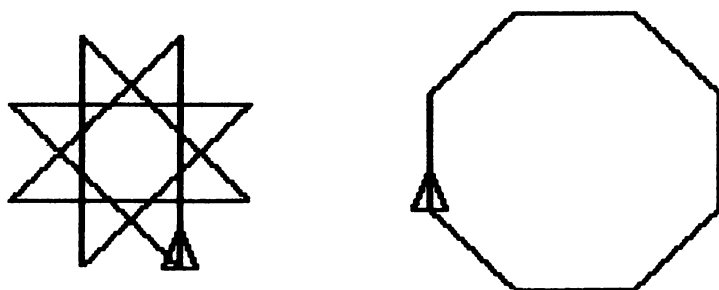



Figura 5.28 Effetto delle procedure STELLA135 e POLIGONO45

Come si farà a tracciare un cerchio con questa tecnica? Per scoprirlo bisogna immedesimarsi nella tartaruga, magari camminando davvero in cerchio con una successione di passetti e di piccole rotazioni.
Attenzione però, che la tartaruga o avanza o ruota, e non può fare le due cose contemporaneamente.
Ecco qui, in Figura 5.29, una parte del percorso seguito dalla tartaruga: è facile completarlo con la tecnica recursiva.

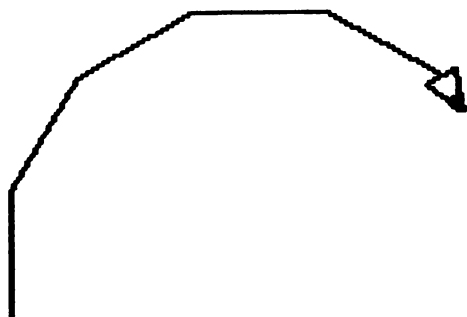
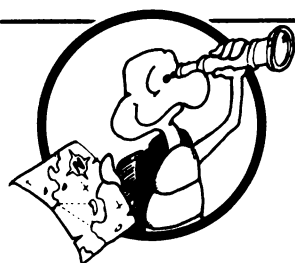



Figura 5.29 Arco di cerchio tracciato con rotazioni di 20 gradi

SUGGERIMENTO

La tecnica recursiva è uno dei cavalli di battaglia del Logo: una procedura che richiama se stessa! Con questa tecnica si riescono a fare in modo semplice cose altrimenti molto complicate.
In questo libro non si vuole insegnare la tecnica recursiva nelle sue espressioni più complesse e impegnative, che riguardano piuttosto il perfezionamento di un programmatore esperto: nel Capitolo 14 sono comunque dati alcuni esempi, che riguardano la struttura delle procedure di servizio speciali.
Se il lettore già conosce il linguaggio BASIC può pensare erroneamente che la tecnica recursiva del Logo corrisponda all'impiego, in BASIC, delle istruzioni GOTO e FOR-NEXT. Bisogna fare attenzione: la tecnica recursiva non va confusa con la tecnica dell'iterazione. Con la tecnica recursiva, infatti, non ci si limita a ripetere istruzioni precedenti, ma si richiama una nuova procedura, con lo stesso nome.
Nel Capitolo 7 verrà ripreso l'argomento, in relazione all'impiego delle variabili e delle condizioni di arresto. Qui la tecnica recursiva è semplicemente un modo per eseguire certi disegni (che sono, da parte loro, un buon esempio di applicazione).

5.5 Archi e cerchi

Ora si può fare qualcosa di più rispetto al Capitolo 2, in cui per la prima volta si sono usate le procedure per tracciare archi e cerchi. Per prima cosa, bisogna leggere il file "CERCHI dal disco delle procedure. Le procedure così recuperate (CERCHIODES, CERCHIOSIN, ARCODES e ARCOSIN) richiedono tutte un numero, che significa la misura del raggio da considerare.

Ecco alcuni esempi; se si nasconde la tartaruga si ottiene proprio l'effetto di Figura 5.30.

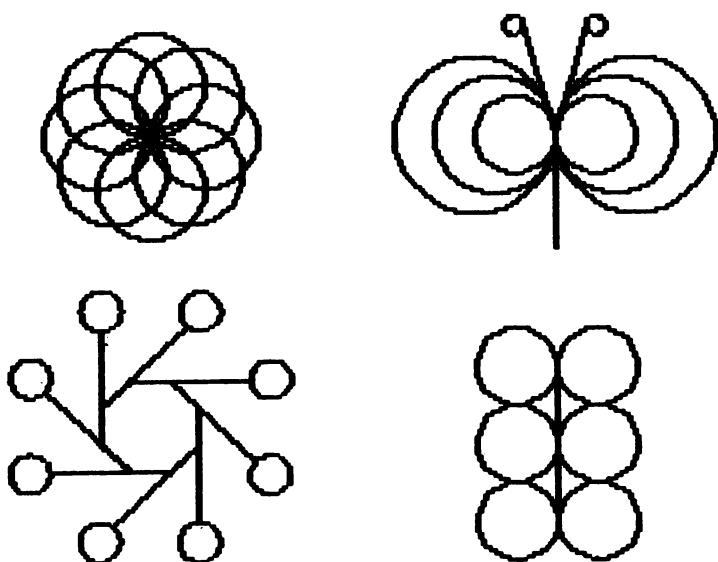



Figura 5.30 Figure tracciate con archi e cerchi

Come si fa il disegno di Figura 5.31?
Basta tracciare un cerchio e poi spostare la tartaruga (con il pennino sollevato), ripetendo quindi l'operazione più volte. Se si fa anche ruotare la

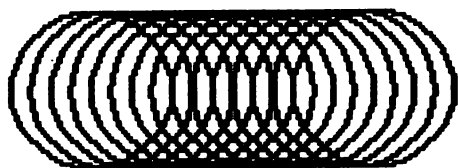



Figura 5.31 Effetto a molla

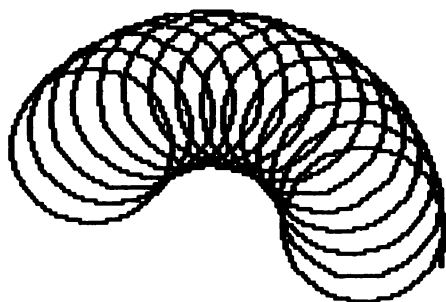



Figura 5.32 Effetto a molla curva

tartaruga, prima di spostarla, si ottiene l'effetto di Figura 5.32.
In Figura 5.33 si vede un altro caso: il simbolo dei giochi olimpici (un consiglio: conviene tracciare separatamente le due righe di cerchi, e poi fare iniziare la seconda riga nel punto adatto, in modo che si intersechi con la prima nel modo voluto).

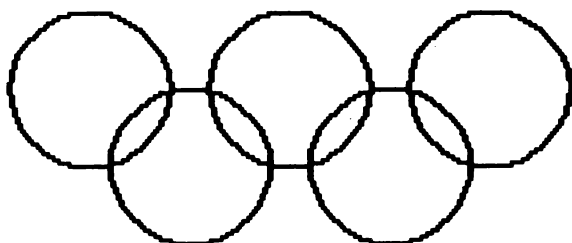



Figura 5.33 Il simbolo delle Olimpiadi

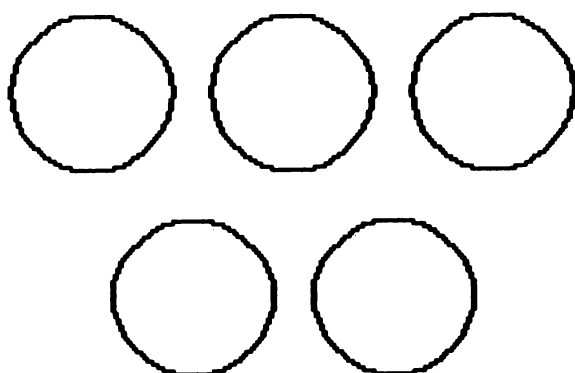



Figura 5.34 Righe di cerchi separate

Ecco ancora un altro caso: come si può tracciare un cerchio partendo con la tartaruga dal centro, e ritornandovi alla fine?
Questo torna molto comodo in diversi casi. In Figura 5.35 si vedono alcuni passi: come si traducono in istruzioni?

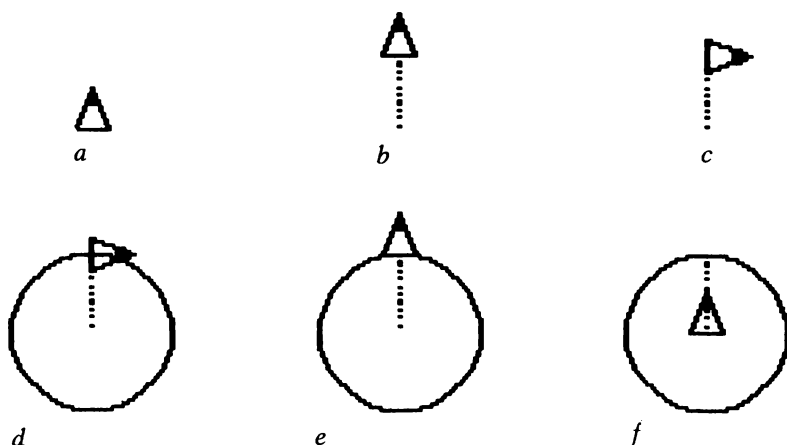



Figura 5.35 Passi per tracciare un cerchio partendo dal centro

In Figura 5.36 si vede come tracciare un altro esempio, una semplice rappresentazione del Sole (qui torna utile saper tracciare un cerchio partendo dal centro).

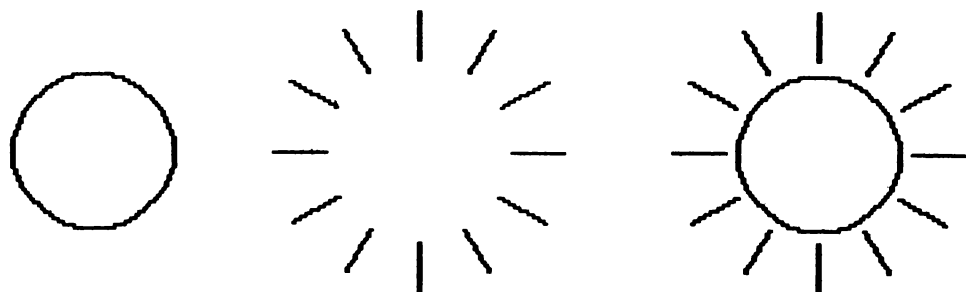



Figura 5.36 Schizzo del Sole formato con un cerchio e con raggi

Le procedure ARCODES e ARCOSIN tracciano quadranti di cerchio (cioè archi ampi 90 gradi): combinandoli, si ottengono effetti interessanti, come in Figura 5.37.
Si può anche tracciare un Sole più elaborato, con un serpentello seguito

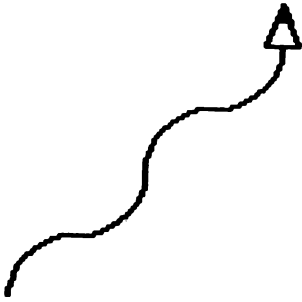



Figura 5.37 Serpentello tracciato con archi di 90 gradi

da un dietro-front quasi completo della tartaruga, ripetendo il tutto molte volte: l'effetto è quello di Figura 5.38.

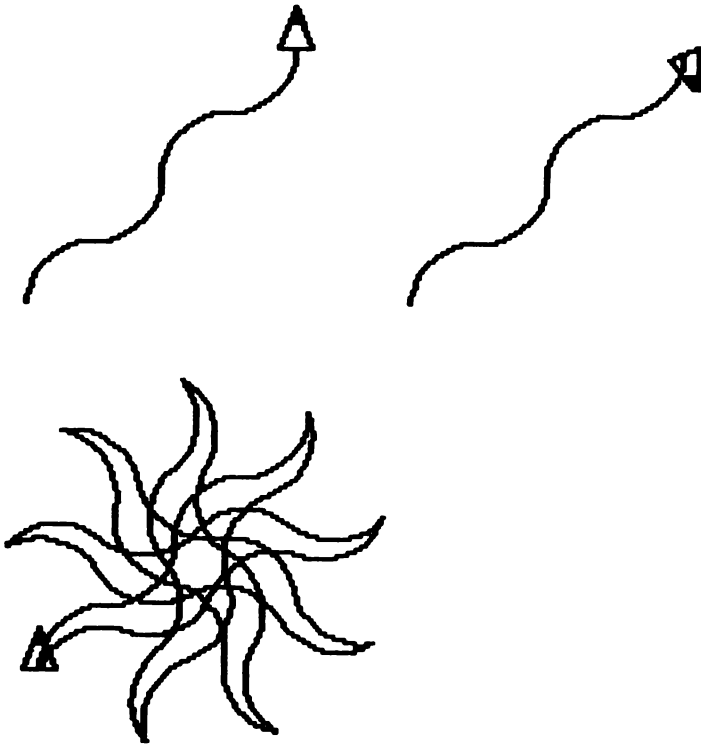



Figura 5.38 Sole con una corona di raggi

Ecco le istruzioni:

```
PER SOLE
BISCIONE
DESTRA 160
SOLE
FINE
```

Si ottengono altri effetti variando l'angolo dell'arco. Ad esempio si può raffigurare un petalo con due archi, come in Figura 5.39.

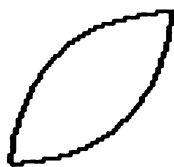



Figura 5.39 Petalo fatto di due archi

Ecco come si traccia il petalo: per prima cosa si decide la misura dell'arco (indicando il raggio):

```
ARCODES 30
```

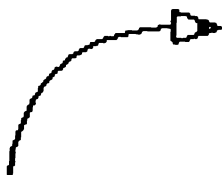



Figura 5.40 Arco tracciato con l'istruzione ARCODES 30

Se si aggiungesse una seconda istruzione identica, si otterrebbe un semicerchio:

```
ARCODES 30
ARCODES 30
```

Per tracciare il petalo, prima di richiamare per la seconda volta la ARCODES, bisogna ruotare la tartaruga. Di quanto? Di 90 gradi: in effetti, per la tartaruga, nel momento della rotazione, il petalo di Figura 5.39 è simile al quadrato, come si vede anche nella costruzione di Figura 5.42.

Figura 5.41 Semicerchio tracciato con due istruzioni ARCODES 30

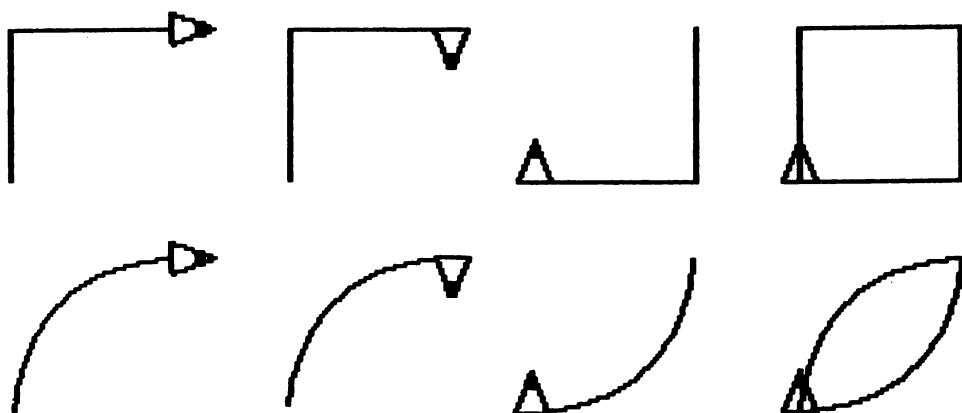



Figura 5.42 Costruire un petalo è un po' come costruire un quadrato

Ecco le istruzioni per il petalo:

```
PER PETALO
ARCODES 30
D 90
ARCOSIN 30
D 90
FINE
```

PETALO riporta la tartaruga al punto di partenza, come QUADRATO già vista nel paragrafo 5.1.
Provate ora a:

- Costruire fiori unendo petali.
- Tracciare una spirale, composta di archi sempre più grandi. Si vedrà più avanti, nel Capitolo 7, come facilitare questo tipo di costruzione, ricorrendo alle cosiddette variabili.

Ecco ancora un'idea su come impiegare gli archi. Si può fare andare la

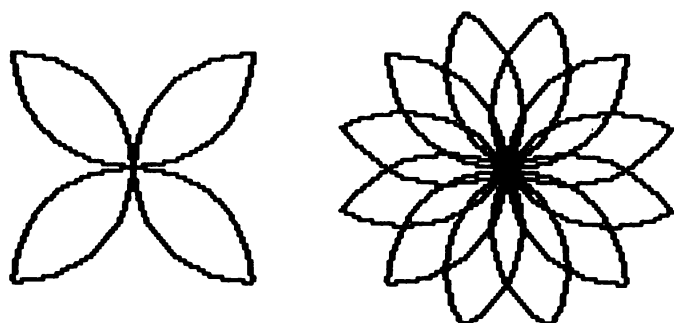



Figura 5.43 Due tipi di fiore

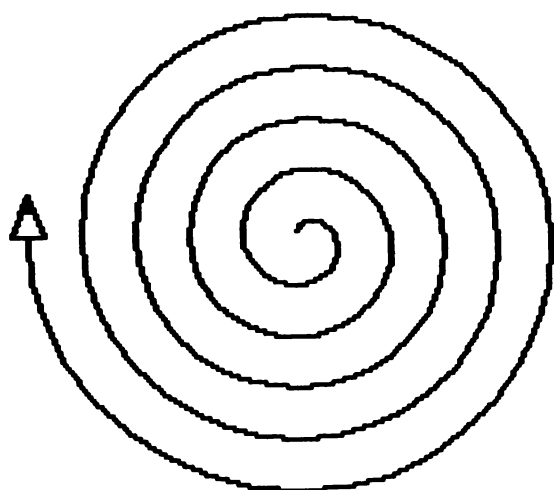



Figura 5.44 Spirale

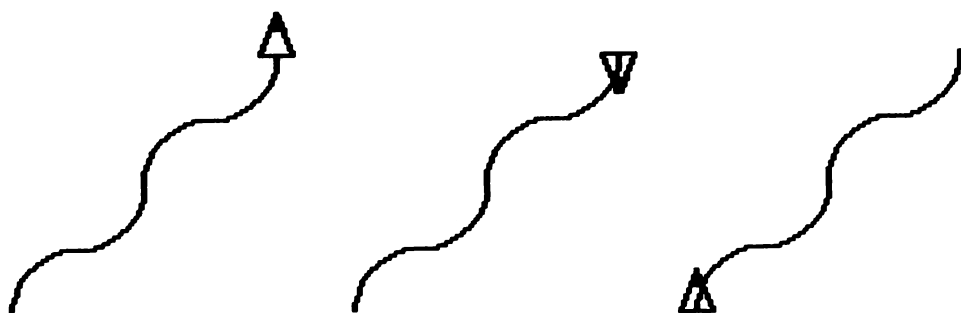



Figura 5.45 La tartaruga può tornare sui suoi passi

tartaruga a ritroso, lungo il percorso già tracciato, usando la stessa procedura:

```
PER BISCIONE
ARCODES 30
ARCOSIN 30
ARCODES 30
ARCOSIN 30
FINE

PER BIBISCIONE
BISCIONE
D 180
BISCIONE
D 180
FINE
```

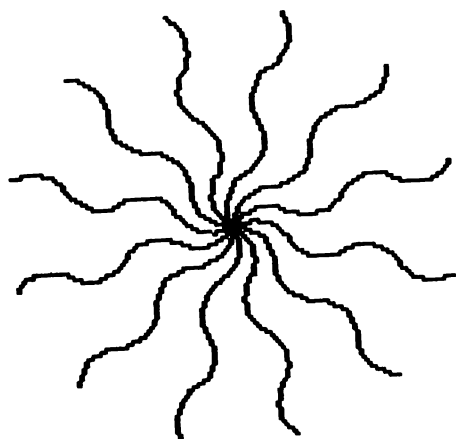



Figura 5.46 Una nuova stella

- Si può combinare BIBISCIONE con la rotazione per disegnare un altro tipo di stella.
- Si può tracciare un'immagine speculare del biscione appena visto, come lo rifletterebbe uno specchio: basta invertire l'ordine con cui si tracciano gli archi verso destra e verso sinistra. Poi si possono combinare le due procedure, BISCIONE e ENOICSIB, per ottenere effetti particolari, come in Figura 5.47.

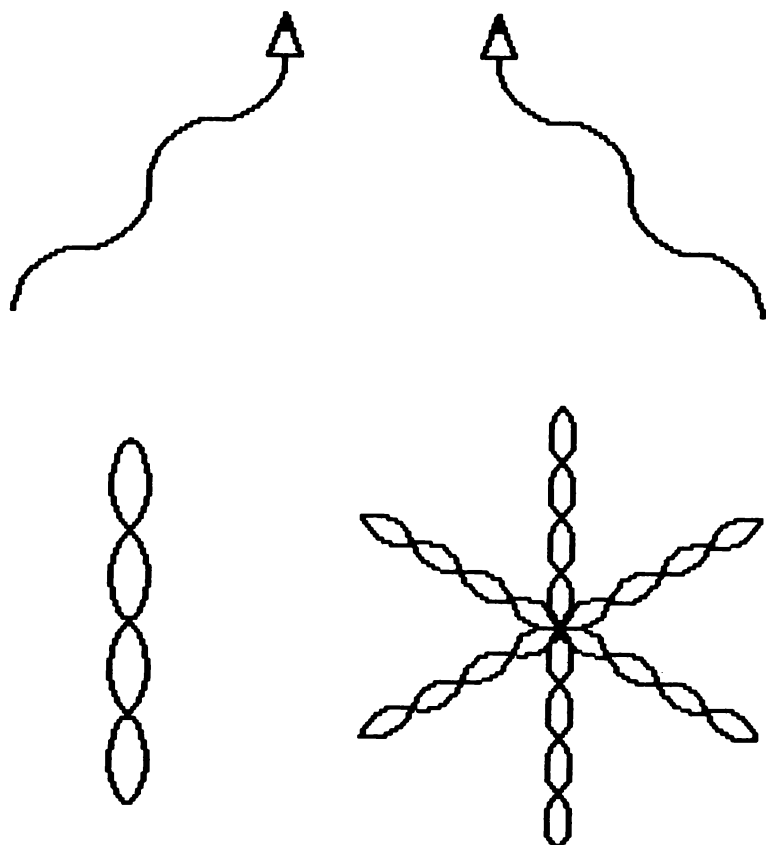



Figura 5.47 Disegni ottenuti con le procedure BISCIONE e ENOICSIB

6

Disegnare con la tartaruga

Con la tartaruga si possono costruire i disegni più vari: persone, animali, automobili, aeroplani, autocarri, fiori, e ogni altra cosa che venga in mente (anche le proprie iniziali, se si vuole).
Con la tartaruga si disegna in modo un po' speciale, perché questa ha bisogno che tutti i movimenti vengano indicati per filo e per segno. Quando però ha imparato un disegno nuovo, non lo scorda più e lo ripete tutte le volte che si vuole.
Non si deve aver paura di disegni nuovi, che a prima vista sembrano difficili: con la tartaruga ogni disegno, anche il più complicato, si scompone in parti molti semplici. Se si fa attenzione, si riesce ad eseguire qualunque disegno, compresi tutti quelli presentati in questo capitolo.
Ecco qui per prima cosa alcuni consigli pratici, che rendono più facile la vita di chi disegna con la tartaruga:

- Disegnare il nuovo schizzo a mano su un foglio di carta, o sul diario del Logo.
- Suddividere il disegno in tante parti elementari, da eseguire una alla volta.
- Dare un nome a ciascuna parte (che verrà richiamato nelle procedure).
- Preparare uno schizzo di insieme, che mostri come si combinano le varie parti del disegno (conviene anche numerarle).
- Semplificare il disegno: se una parte è troppo complicata, conviene eliminarla o sostituirla con una più semplice.
- Le procedure possono essere scritte separatamente, per le singole parti, che si uniscono in un secondo tempo. Se si preferisce, si possono anche inserire le nuove parti nel disegno complessivo appena sono pronte, per vedere subito come si adattano.

SUGGERIMENTO

Lo sviluppo delle procedure è un fatto di gusto e di preferenze personali, per cui non esiste un metodo unico da seguire. Nel libro si segue l'approccio detto top-down, con il quale si parte dalla struttura generale e man mano si scende nei dettagli delle singole parti. Questo modo di fare è adatto a chi riesce a ragionare facilmente con concetti astratti; molti comunque sviluppano gradualmente un loro metodo personale, provando e riprovando. Le idee qui presentate su come si sviluppano i programmi hanno valore di suggerimenti, di indicazioni da provare, e non vanno imposte come se fossero verità assolute.
L'esperienza fatta nell'insegnamento del Logo dimostra che molti allievi vanno lasciati liberi di scegliere il metodo di sviluppo che preferiscono, anche se ciò comporta necessariamente un periodo di prove e di errori. Per chi aiuta un principiante, occorre qui molta pazienza anche per cercare di capire come ciascuno intende il proprio modo di fare: solo così si potranno dare consigli veramente utili. Imporre a tutti un solo modo di sviluppare i programmi sarebbe assurdo e dannoso, come lo sarebbe l'imposizione di un unico progetto da sviluppare. Chi impara, dopo un certo tempo, raggiunge una propria maturità e padroneggia sia il linguaggio che il modo di usarlo: questa è però una conquista personale, che non può essere fatta per procura, o per imposizione dall'esterno.
In fondo, l'essenza dell'insegnamento consiste nel saper consigliare e nel saper tacere e lasciar fare, a tempo debito.
Più ancora, forse, ci vuole umiltà, per dare un consiglio e accettare di vederlo magari respinto da chi impara.
Bisogna ancora osservare che non è da tutti saper vedere un progetto da più punti di vista contemporaneamente, come insieme di parti e come entità autonoma. Molti rifiutano, in pratica, di impiegare le procedure secondarie e partono senz'altro con il disegno completo, che cercano di realizzare in modo intuitivo.
Comunque, quando si è completata una parte del disegno, o si è arrivati a buon punto, conviene registrare ciò che si è fatto, con il nome che si preferisce: questo accorgimento pratico può anche essere un buon modo di presentare all'allievo il principio della realizzazione separata delle parti.
La suddivisione intenzionale di un disegno in singole parti da realizzare separatamente viene capita e accettata, in molti casi, solo dopo un certo tempo.

6.1 Come disegnare un camion

Supponiamo di voler disegnare con la tartaruga un camion, come quello riportato nello schizzo di Figura 6.1.

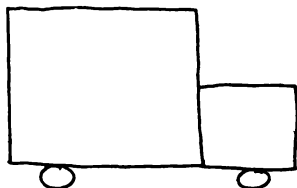



Figura 6.1 Schizzo di un camion

Si può pensarlo composto di tre parti: la cabina di guida, il cassone e le ruote.
Le parti possono poi essere combinate come si vuole (è un bel vantaggio).
Cominciamo con ciascuna parte, per conto suo, ripulendo ogni volta lo schermo e partendo sempre con la tartaruga in posizione tana:

```
PER CASSONE
RIPETI 4 [AVANTI 60 DESTRA 90]
FINE

PER CABINA
RIPETI 4 [AVANTI 30 DESTRA 90]
FINE

PER RUOTE
DESTRA 90
CERCHIODES 5
AVANTI 90
CERCHIODES 5
INDIETRO 90
SINISTRA 90
FINE
```

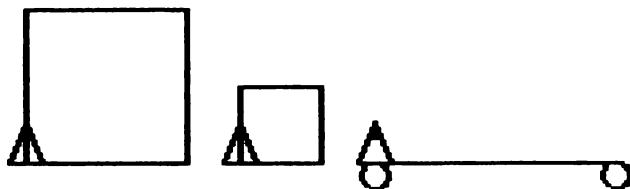



Figura 6.2 Le parti del camion, ancora separate

Per tracciare le ruote, occorre leggere da un disco il file "CERCHI. Se non l'avete già fatto, dovete ricopiare la procedura CERCHIODES dall'Appendice A.

TRABOCCHETTO

Fare attenzione di inserire nel drive il disco di servizio che si usa normalmente, dopo aver letto dal disco il file "CERCHI.

E ora si monta il camion:

```
PER CAMION
CASSONE
CABINA
RUOTE
FINE
```

Battendo infine il nuovo comando, CAMION, si ottiene l'effetto di Figura 6.3.

TRABOCCHETTO

Eh No! In Figura 6.3 non si vede un camion! Non bisogna scoraggiarsi, anzi. Chi usa un computer deve abituarsi a questo tipo di errore: l'importante è trovare le correzioni giuste.
In questo caso, l'errore è banale, perché le parti del camion vanno semplicemente disposte in modo diverso.

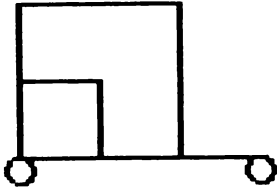
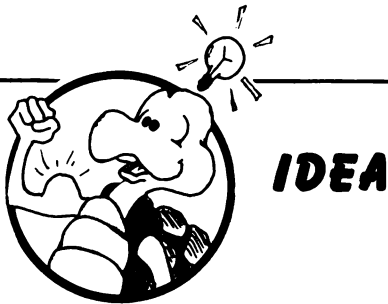



Figura 6.3 Primo tentativo di montare il camion

Occorrono qui due nuove procedure, per spostare le parti avanti e indietro: si possono chiamare, ad esempio, PARTAVAN e PARTINDIE. Per scriverle, ci sono molti modi: conviene pensarci su e trovarne uno da soli (è un buon allenamento).

Gli errori sono praticamente inevitabili, usando il computer. Un po' è colpa nostra, ma anche il computer fa la sua parte, perché è veramente privo di fantasia e fa sempre esattamente quello che gli viene detto: va condotto per mano.

La Figura 6.4 illustra l'effetto delle due nuove procedure da realizzare, PARTAVAN e PARTINDIE.

Figura 6.4 Effetto da realizzare con le procedure PARTAVAN e PARTINDIE

Ecco le istruzioni (è uno dei possibili modi):

```
PER PARTAVAN
D 90
A 60
S 90
FINE
PER PARTINDIE
S 90
A 60
D 90
FINE
```

A questo punto si può modificare la procedura CAMION, inserendovi le due nuove procedure:

```
PER CAMION
CASSONE
PARTAVAN
CABINA
PARTINDIE
RUOTE
FINE
```

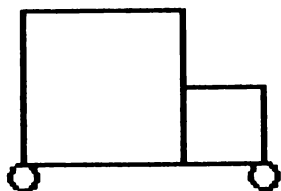



Figura 6.5 Il camion montato, al secondo tentativo

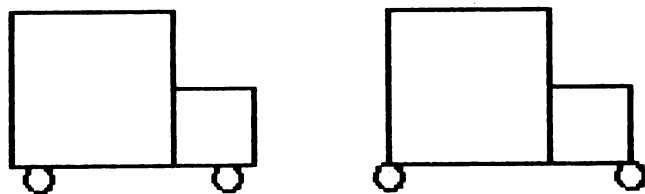



Figura 6.6 Due esempi di camion: quello di sinistra è più simile all'originale

Finalmente! Ci sarebbe ancora un particolare, però: le ruote vanno un po' più all'interno. A questo punto, le modifiche non sono più un mistero e si può fare da soli, spotando le ruote dove si preferisce (con la procedura RUOTE).

Ci si può dimenticare di registrare il lavoro fatto. Attenzione. Qualunque nome, come al solito, va bene, ad esempio "CAMION. Prima di passare ad altro, conviene allora battere CONSERVA "CAMION.

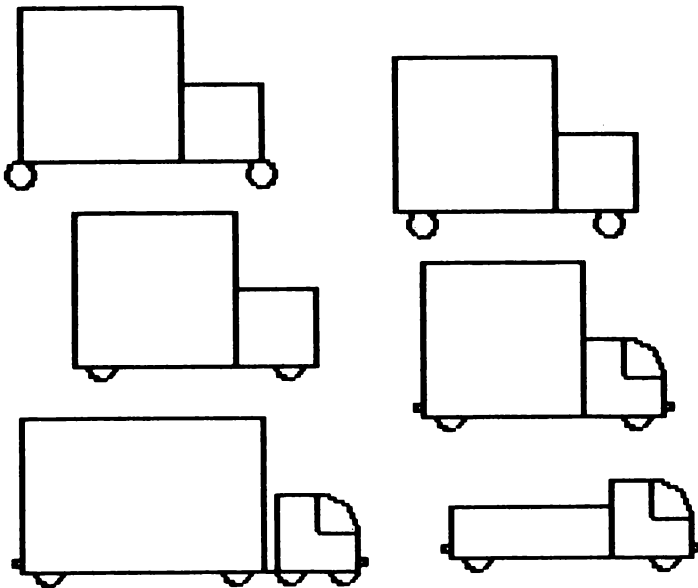
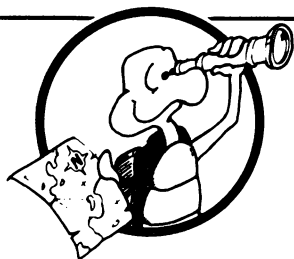



Figura 6.7 Un parcheggio di Logocamion

ESPLORAZIONE

Ora che si è montato il camion, si possono aggiungere altri particolari, o costruire nuovi modelli. Si possono aggiungere i paraurti, o i vetri della cabina, o modificare il modello di partenza per ottenere un autotreno, o un furgone. In Figura 6.7 si vede un parcheggio di Logocamion.

SUGGERIMENTO

Nell'esempio appena visto si è commesso di proposito un errore iniziale per presentare due concetti importanti. Primo, gli errori sono sempre possibili (e talvolta sono anche interessanti e persino divertenti). Secondo, non sempre gli errori sono dannosi, anzi a volte conducono ad un risultato utile, anche se inaspettato. In effetti, non è detto che si debba sempre eseguire alla lettera il progetto di partenza.

Il primo errore dell'esempio di costruzione del camion riguarda la posizione e l'orientamento della tartaruga, che non sono stati tenuti nel giusto conto durante la costruzione delle singole parti. Si è ignorata infatti l'esigenza di spostare la tartaruga dopo aver tracciato il cassone, prima di tracciare la cabina. Questo tipo di errore è molto frequente, in tutti i tipi di costruzione grafica.

Il secondo errore è dovuto ad una semplificazione nella costruzione di una parte (le ruote), che spesso è utile per rendere più spedita la preparazione di una prima versione del disegno completo: in genere, c'è modo di aggiungere nuovi particolari in seguito, dopo aver visto l'effetto complessivo del disegno.

È importante infine tener sempre presente che un progetto è una guida, non una legge irrevocabile. Conviene anzi essere molto flessibili e ammettere cambiamenti del progetto iniziale, in ogni fase di realizzazione. Se si incontra una difficoltà troppo grande, non bisogna esitare a semplificare il problema (rimandando magari ad un momento successivo la soluzione della difficoltà che al momento non si può superare). E poi, se una buona idea viene in ritardo, per rendere più interessante il progetto, meglio tardi che mai!

6.2 Disegnare le persone

Ecco un altro esempio di costruzione grafica, anch'esso facile: lo schizzo di una persona. In Figura 6.8 se ne vede un esempio.

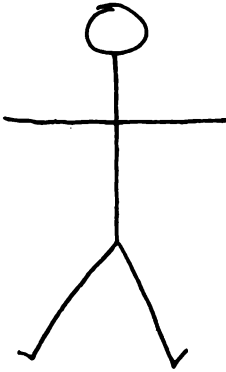



Figura 6.8 Schizzo a mano libera di persona

Queste sono le istruzioni:

```
PER OMINO
TRONCO
GAMBE
BRACCIA
TESTA
CAPPELLO
FINE
```

OMINO è una superprocedura, che cioè controlla l'esecuzione di altre procedure e realizza la costruzione grafica completa. È come il capo di un progetto, che dirige l'attività di un certo numero di collaboratori. Qui basta richiamare la OMINO, che pensa a tutto, richiamando a sua volta i suoi aiutanti.
Se però si batte il nuovo comando OMINO senza aver insegnato al computer come eseguire i comandi TRONCO, GAMBE, BRACCIA, ecc., il Logo protesta (giustamente), come si vede se ci si prova.
Occorre assegnare le istruzioni per eseguire le procedure richiamate da OMINO. Ecco il tronco:

```
PER TRONCO
AVANTI 30
```

```
INDIETRO 30
FINE
```

Figura 6.9 Ecco cosa traccia la procedura TRONCO

Le gambe sono fra loro simmetriche, cioè si rispecchiano l'una con l'altra. Si costruisce la procedura GAMBE con due procedure di rango inferiore, GAMBADES e GAMBASIN, tenendo conto della simmetria appena osservata:

```
PER GAMBE
GAMBASIN
GAMBADES
FINE
```

La GAMBASIN fa ritornare la tartaruga al punto di partenza, da dove può poi iniziare la GAMBADES:

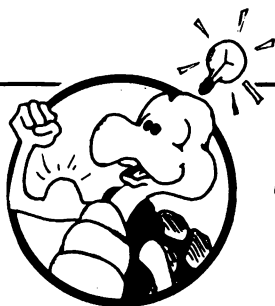
```
PER GAMBASIN
DESTRA 30
INDIETRO 30
SINISTRA 90
AVANTI 5
INDIETRO 5
DESTRA 90
AVANTI 30
SINISTRA 30
FINE
```

Figura 6.10 Effetto della procedura GAMBASIN

Gli ultimi quattro passi, come si vede, riproducono i primi passi in modo speculare, così da riportare la tartaruga al punto di partenza. La procedura GAMBADES è simile alla GAMBASIN, ma ha tutte le rotazioni in senso opposto:

```
PER GAMBADES
S 30
I 30
D 90
A 5
I 5
S 90
A 30
D 30
FINE
```

La Figura 6.11 mostra l'effetto del nuovo comando GAMBE.

Figura 6.11 Effetto della procedura GAMBE

IDEA

Quando una figura è simmetrica, il lato sinistro è l'immagine speculare del lato destro, e l'alto del basso. La simmetria è importante, perché può far risparmiare molto lavoro. Ad esempio, per un disegno con simmetria destra/sinistra, basta costruire una sola parte; per l'altra è sufficiente ripetere la costruzione, invertendo il senso di tutti gli angoli.
Bisogna però avere l'avvertenza di far terminare la tartaruga, nella prima parte, al punto da cui è partita.

Se ora si batte il comando OMINO, il computer traccia parte della figura in progetto e stampa un messaggio in cui è detto che mancano le istruzioni della procedura BRACCIA.

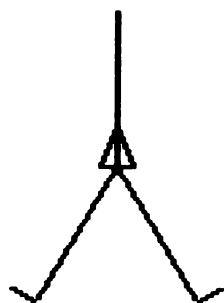



Figura 6.12 GAMBE e TRONCO: effetto congiunto

Ecco le istruzioni per la procedura BRACCIA:

```
PER BRACCIA
A 20
D 90
A 20
I 40
A 20
S 90
I 20
FINE
```

Figura 6.13 Effetto della procedura BRACCIA

Battendo ora il comando OMINO, si dovrebbe ottenere l'effetto di Figura 6.14.
Lo schizzo della testa, che ancora manca, è simile a un palloncino, come si vede in Figura 6.15.
Potete provare a scrivere da soli le istruzioni per tracciare la testa dell'omino (conviene far tornare la tartaruga al punto di partenza in modo da facilitare la connessione delle varie parti).

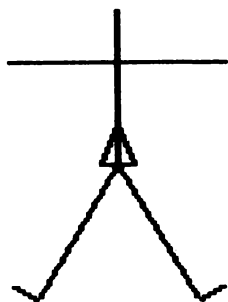



Figura 6.14 L'omino ha bisogno di una testa

Figura 6.15 Questo palloncino potrebbe andare bene come testa per l'omino

Ed ecco, infine, le istruzioni per il cappello:

```
PER CAPPELLO
SULAPENNA
A 40
GIULAPENNA
BERRETTO
SULAPENNA
I 40
FINE
```

Come si vede, con CAPPELLO si richiama un'altra procedura, di nome BERRETTO, che traccia materialmente lo schizzo del cappello, facendo uso della ARCODES (chiamata due volte):

```
PER BERRETTO
S 90
A 10
I 5
D 90
ARCODES 5
```

```
ARCODES 5
S 90
A 5
I 10
S 90
FINE
```

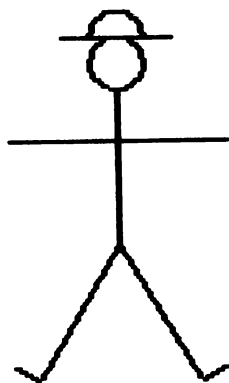



Figura 6.16 L'omino completo

Se si sceglie un raggio diverso da 5 (il valore di esempio) per il cappello, bisogna naturalmente modificare anche il primo e l'ultimo passo della procedura CAPPELLO, in modo che lo schizzo risulti bene. Ecco una domanda di prova: come si deve modificare la CAPPELLO in modo da ottenere l'effetto di Figura 6.17?

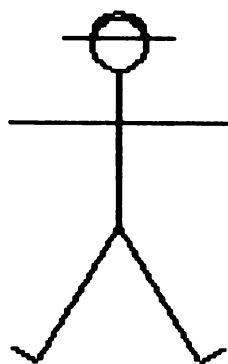
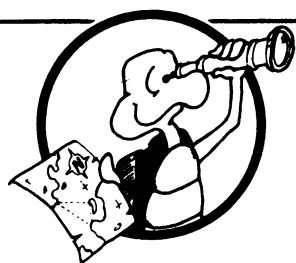



Figura 6.17 Omino col cappello calcato

ESPLORAZIONE

Le variazioni possibili sono moltissime (ci si può davvero sbizzarrire, come si vede in Figura 6.18).

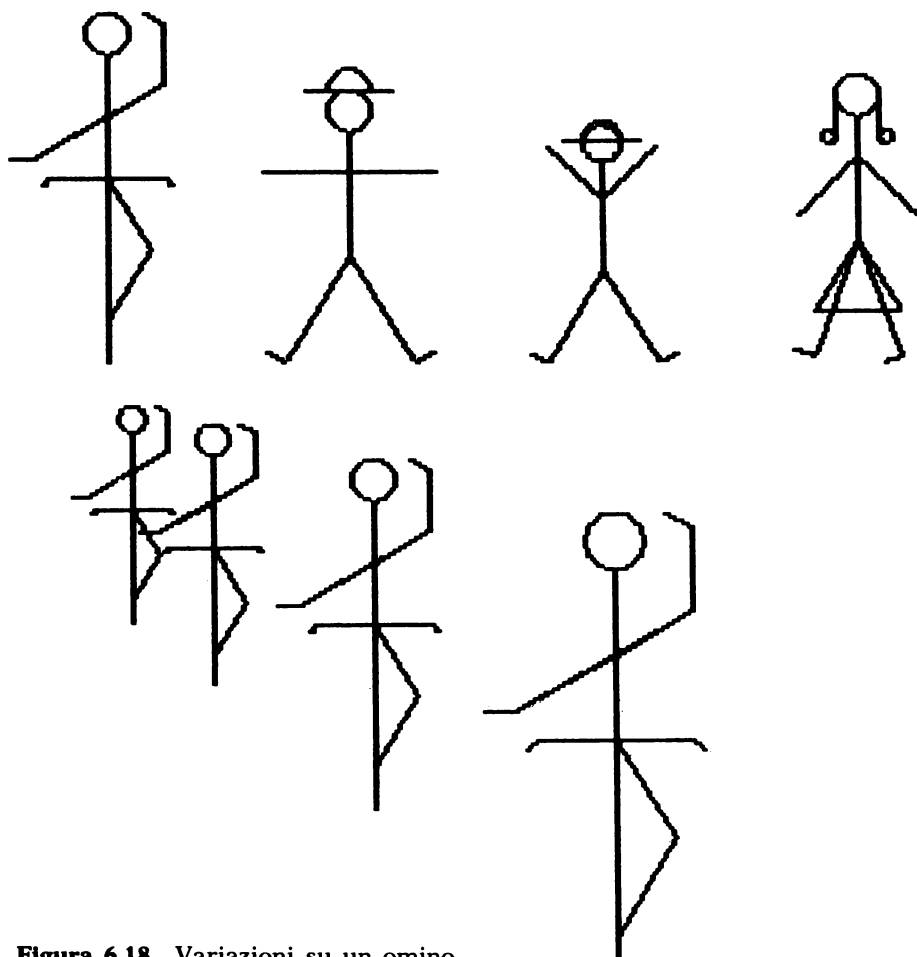



Figura 6.18 Variazioni su un omino

SUGGERIMENTO

Nell'esempio appena visto sono presentati diversi concetti importanti:

1. La simmetria è una proprietà notevole e anche una buona occasione di semplificare la costruzione pratica di certi disegni. La simmetria è anche un esempio di come convenga suddividere in parti la realizzazione di un progetto, perché in tal modo se ne riduce la difficoltà.
2. Una procedura in cui la tartaruga, al termine, si ritrova al punto di partenza (e con lo stesso orientamento con cui è partita) è particolarmente adatta a combinarsi con altre. In effetti, in questo modo la procedura è, come anche si dice, trasparente, nel senso che non importa l'ordine con cui è richiamata, rispetto ad altre procedure. Ad esempio, si potrebbe scrivere la procedura OMINO in ordine inverso senza cambiare l'effetto finale, proprio perché è composta di procedure trasparenti, nel senso chiarito prima. Una procedura trasparente è anche sostituibile con molta facilità, perché non ci si deve preoccupare dell'effetto sulle altre procedure (a patto che anche la nuova procedura sia trasparente, cioè restituisca la tartaruga nello stesso stato in cui la trova). Nel caso dell'omino, è molto semplice cambiare la posizione, ad esempio, delle braccia: basta modificare la sola procedura BRACCIA, e niente altro.
3. La scomposizione intenzionale di un progetto in parti più semplici è un metodo di lavoro da tener sempre presente. OMINO è sia una superprocedura (nel senso che realizza l'intero disegno dell'omino) che il progetto della costruzione grafica dell'omino stesso. Le istruzioni della procedura OMINO possono valere anche da guida per la realizzazione del progetto, in generale. Addirittura, se si assegna la OMINO al computer prima di avere preparato le procedure in essa richiamate, si realizza in pratica un piano di lavoro che può essere verificato man mano che la scrittura delle varie parti viene completata. Certo, come ogni piano di lavoro, anche la OMINO è modificabile, se viene un'idea migliore in corso d'opera (nel caso proposto, invece, si è realizzato esattamente il piano originale).

6.3 Il disegno di un fiore

Ci sono molti modi di disegnare un fiore. Qui se ne vede uno che impiega archi di cerchio per tracciare le foglie e i petali. L'elemento usato è la procedura chiamata PETALO, che richiama le procedure ARCODES e ARCOSIN.

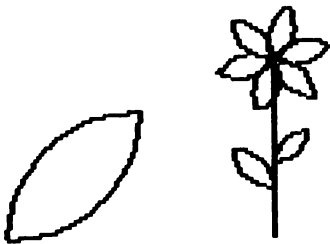
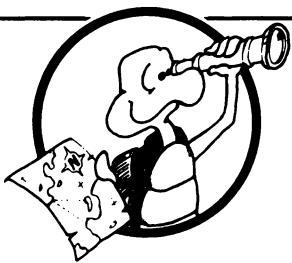



Figura 6.19 Un petalo e un fiore

Come si disegna un petalo? Basta ricordarsi del paragrafo 5.5, dove si è visto proprio questo esempio. Bisogna naturalmente leggere dal disco il file "CERCHI, oppure ricopiare le procedure che servono dall'Appendice A. Conviene preparare subito diversi petali, grandi e piccoli, da usare di volta in volta per costruire una grande varietà di fiori.

ESPLORAZIONE

- Disegnare diverse corolle, richiamando più volte la procedura PETALO (se ne possono fare di grandi e di piccole, e di varia forma, come in Figura 6.20).
- Disegnare un fiore completo, con la corolla, lo stelo e le foglie.
- Disegnare un giardino, con tanti tipi di fiori (colorati, se si ha uno schermo a colori).

Si vedrà più avanti, nel Capitolo 7, il modo di preparare procedure per

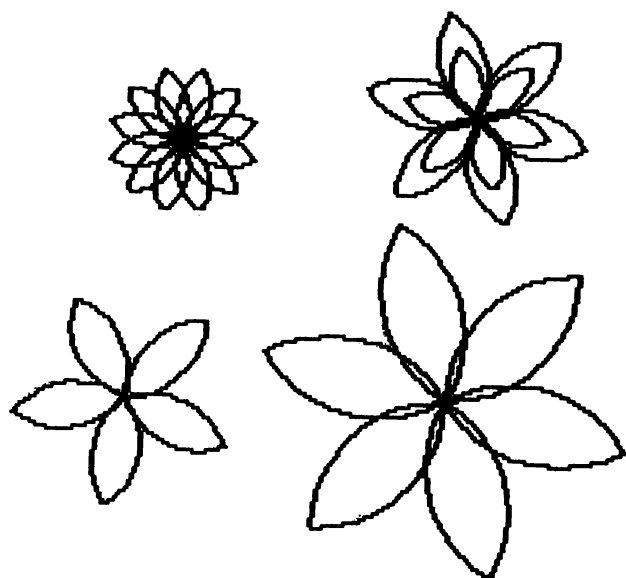



Figura 6.20 Corolle

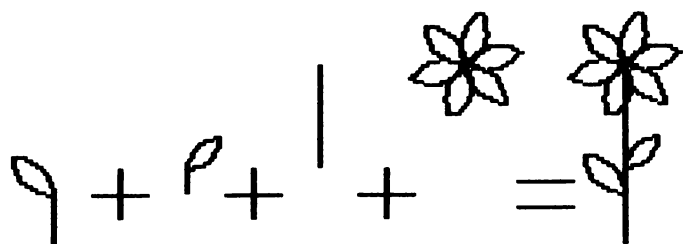



Figura 6.21 La costruzione del fiore

costruire figure di varia dimensione, secondo il numero di volta in volta battuto alla tastiera. Sarà allora più facile disegnare un giardino veramente vario.

6.4 Altre idee

Qui sono riportate le costruzioni grafiche più riuscite, ideate da allievi nel corso dell'apprendimento del Logo. Possono servire come esempio di cosa si può fare con il Logo.

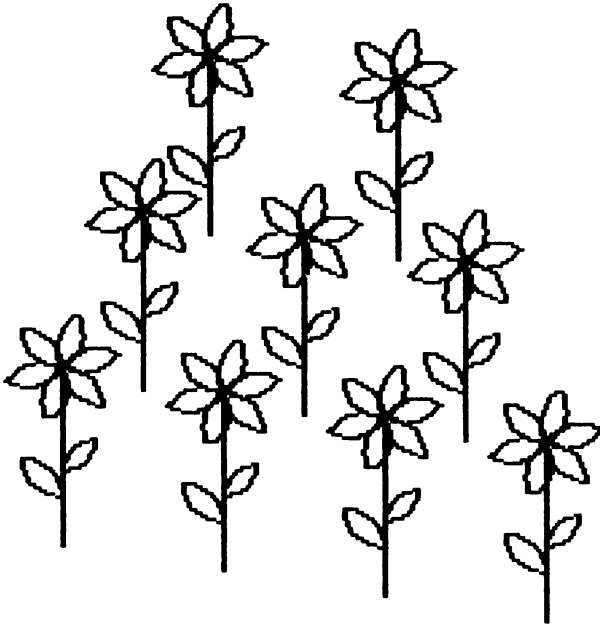
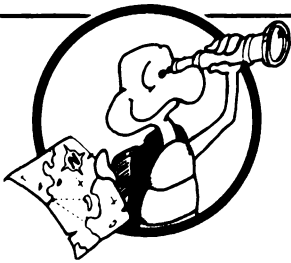



Figura 6.22 Ecco un Logogiardino

FACCE

Le facce sono simmetriche (almeno, a prima vista) e si può fare buon uso di questa proprietà per semplificare la realizzazione dei disegni. In certi casi, però, gli effetti più interessanti si ottengono con una faccia non simmetrica (almeno in certe parti).
Si riesce a capire come sono state costruite le facce di Figura 6.23?
In Figura 6.24 si vede una faccia realizzata da un allievo di 11 anni. Egli incominciò con una superprocedura, di nome TESTA, realizzando quindi, nell'ordine in cui si presentano, le procedure secondarie. Il lavoro fu lungo e ci volle parecchia pazienza (alcune procedure secondarie, a loro volta, ne richiamano altre ancora).

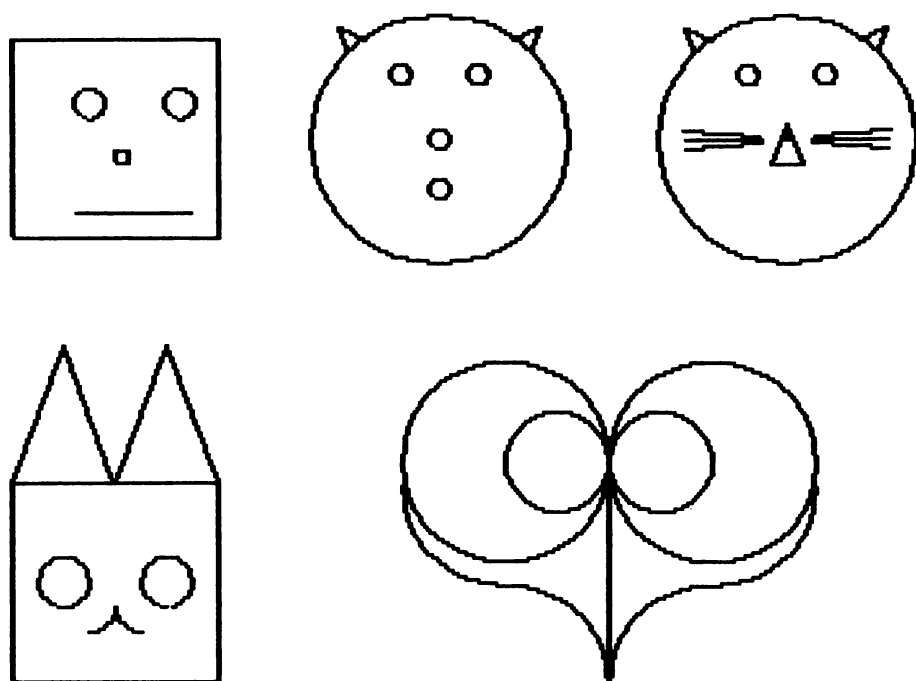



Figura 6.23 Facce

Ecco la procedura:

```
PER TESTA
QUADRATO
OCCHI
NASO
BOCCA
BARBA
CAPELLI
ORECCHI
CAPPELLO
FIORE
FINE
```

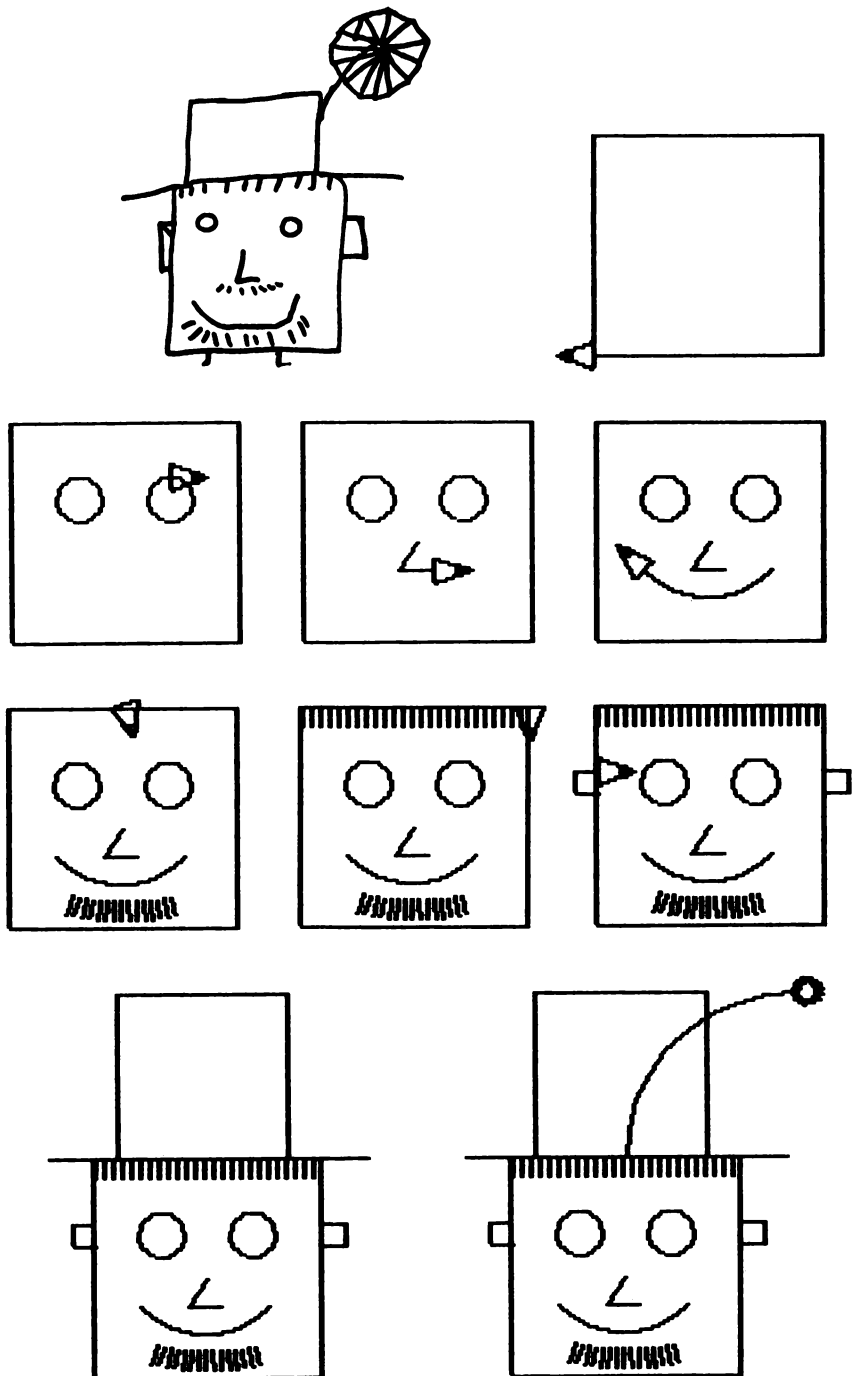



Figura 6.24 La faccia nelle varie fasi di costruzione

ANIMALI

Con la tartaruga, si riesce a fare schizzi molto rappresentativi di ogni genere di animale: farfalle, pesci, cani, gatti, ecc.

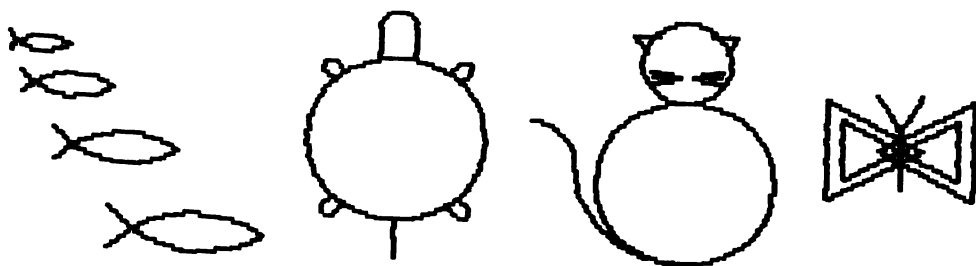



Figura 6.25 Ecco un Logozoo

VEICOLI

In Figura 6.26, si vedono alcune rappresentazioni molto popolari fra i piccoli allievi, che riguardano i veicoli più disparati.

BASEBALL

In Figura 6.27 si vede cos'hanno fatto due piccoli allievi, decidendo insieme il piano di lavoro e realizzando le parti separatamente. Il progetto faceva parte di un progetto ancora più ambizioso, che doveva realizzare la simulazione del gioco del baseball con il Logo, ma che non fu mai portato a termine. I due ragazzi, comunque, si divertirono un mondo a fare la loro parte.

ALBERI

In Figura 6.28 si vede come, da un rametto, può nascere un intero bosco.

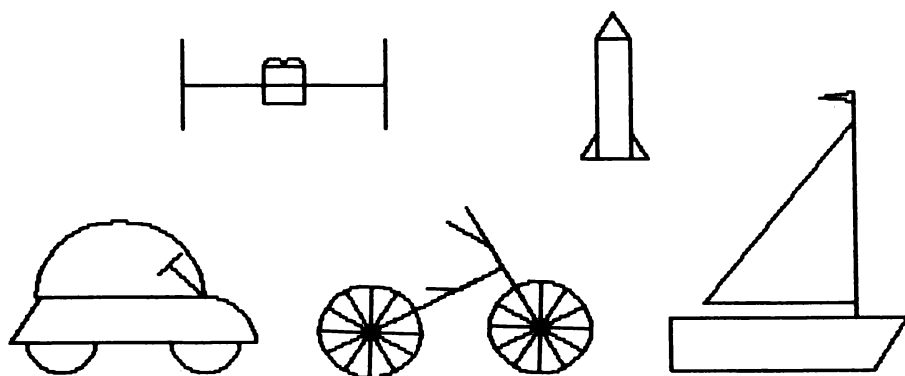



Figura 6.26 Dalla bicicletta all'astronave

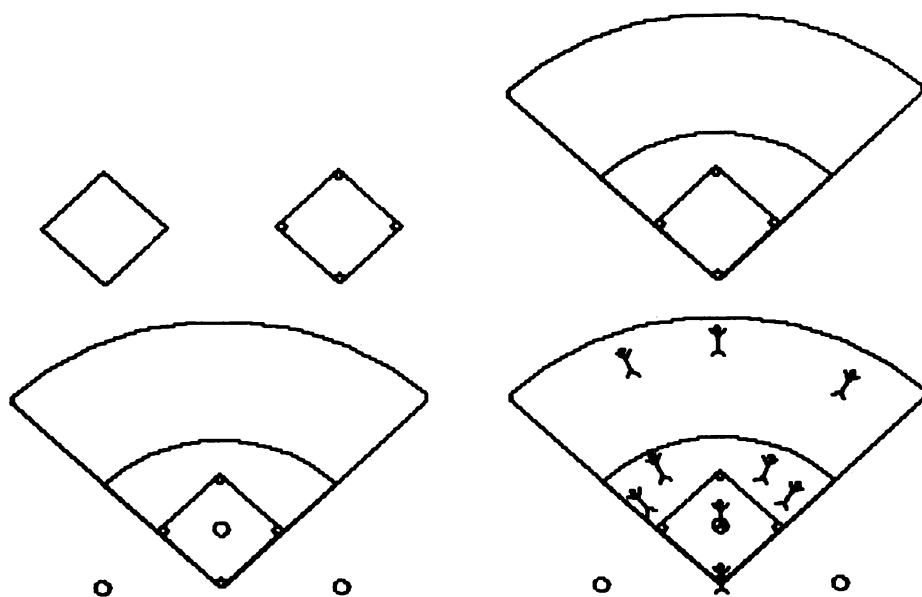



Figura 6.27 Costruzione di un campo di baseball

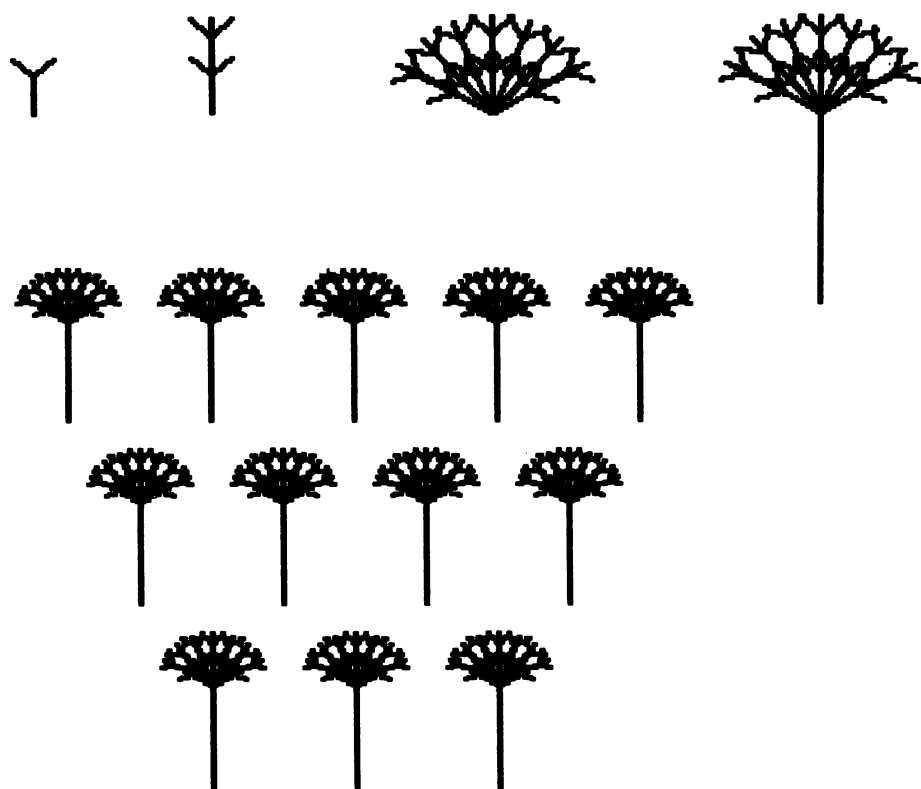



Figura 6.28 Ramo, chioma, albero, bosco

7

Le variabili

Questo capitolo presenta le cosiddette variabili, con le quali il Logo memorizza le informazioni che vengono man mano fornite da chi lo usa. Una variabile è un elemento di informazione, con un nome che lo distingue da tutte le altre variabili. Il computer memorizza, nella sua memoria di servizio, sia il nome che il valore dell'informazione: questo valore, poi, può essere modificato in qualunque momento (ecco perché si parla di variabili).
Le variabili servono per memorizzare qualunque genere di informazione: il Logo memorizza numeri, parole e stringhe (si vedranno più avanti, nel Capitolo 9; per ora si usano solo numeri).
Ci sono due modi di creare una variabile: nel Capitolo 9 si vedrà come usare il comando ASSEGNA (che è un comando originale del Logo) con questo scopo; qui si creano le variabili richiamando procedure che richiedono l'assegnazione di numeri. Anche le procedure (come QUADRATO, o TRIANGOLO) possono avere dati numerici, battuti di volta in volta alla tastiera, proprio come i comandi del Logo (ad esempio, AVANTI, DESTRA, ecc.) e le variabili sono il modo in cui le procedure memorizzano questi dati.

Comando	**Abbreviazione**	**Esempi**
+		AVANTI :GRANDEZZA+10 STAMPA 5+3
SE		SE :GRANDEZZA<10 STOP
STOP		SE :GRANDEZZA<10 STOP

Comando	Abbreviazione	Esempi
<		SE :GRANDEZZA<10 STOP
>		SE :ANGOLO>90 STOP
=		SE :GRANDEZZA = 100 STOP STAMPA 5 = 3+2
DIREZIONE		IF DIREZIONE = 0 STOP STAMPA DIREZIONE

Viene usato il file di procedure "CERCHI.
Non vengono introdotte nuove procedure di servizio.

7.1 Dati che cambiano le dimensioni di un disegno

In Figura 7.1 si vede un disegno formato da cinque quadrati, uno interno all'altro. Si potrebbe disegnare, senza ricorrere all'uso di variabili, co-

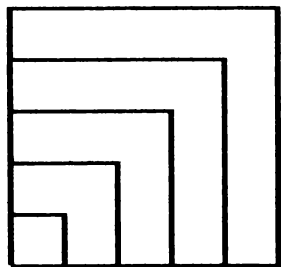



Figura 7.1 Disegno con cinque quadrati

struendo un quadrato alla volta e indicando man mano gli spostamenti da eseguire con il comando AVANTI. Ricorrendo alle variabili si può scrivere invece una sola procedura, che disegna quadrati di varia grandezza, a seconda del numero che si indica di volta in volta.
Ecco la procedura che impiega le variabili:

```
PER QUADRATO :GRANDEZZA
AVANTI :GRANDEZZA
DESTRA 90
AVANTI :GRANDEZZA
DESTRA 90
AVANTI :GRANDEZZA
```

```
DESTRA 90
FINE
```

Ecco un modo alternativo:

```
PER QUADRATO :GRANDEZZA
RIPETI 4 [AVANTI :GRANDEZZA DESTRA 90]
FINE
```

Al posto di :GRANDEZZA, si può usare qualunque altra parola (è sempre meglio scegliere nomi che ricordano la funzione della variabile; anche i nomi delle procedure, come ad esempio QUADRATO, vanno del resto scelti con lo stesso criterio).
Allora, per far tracciare un quadrato alla tartaruga, si batte il comando QUADRATO, seguito da un numero che viene interpretato come la lunghezza del lato (proprio come si fa con il comando AVANTI). Ecco un esempio:

```
QUADRATO 30
QUADRATO 40
QUADRATO 50
ecc.
```

Per realizzare il disegno di Figura 7.1 basta allora scrivere questa semplice procedura:

```
PER RIQUADRO
QUADRATO 10
QUADRATO 20
QUADRATO 30
QUADRATO 40
QUADRATO 50
FINE
```

TRABOCCHETTO

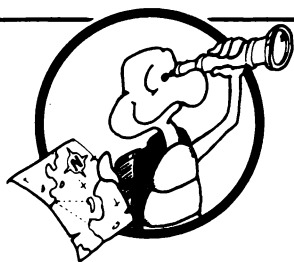
Fare attenzione ai due punti ":", che compaiono prima dei nomi delle variabili, come in :GRANDEZZA. Ecco le regole da tenere presente:

1. Non lasciare spazi fra i due punti e il nome. Se si scrive, ad esempio, : GRANDEZZA, il Logo protesta.
2. Quando si batte il valore di una variabile non vanno usati i due punti. Ecco cosa succede se invece si battono i due punti anche prima del valore:

```
QUADRATO :30
MANCA IL NOME 30
```

Il Logo interpreta ciò che segue immediatamente i due punti come il nome di una variabile. Qui, ad esempio, ricerca la variabile chiamata 30, e non la trova (perciò protesta).

Le variabili si applicano bene a molti casi pratici, come ad esempio al disegno di stelle e triangoli (già visti nel Capitolo 5).
Ecco due casi:

```
PER STELLA :GRANDEZZA
RIPETI R [AVANTI :GRANDEZZA DESTRA 144]
FINE

PER TRIANGOLO :GRANDEZZA
RIPETI 3 [AVANTI :GRANDEZZA DESTRA 120]
FINE
```

ESPLORAZIONE

Conviene provare qualche disegno di fantasia, composto magari con stelle e triangoli di varia grandezza.

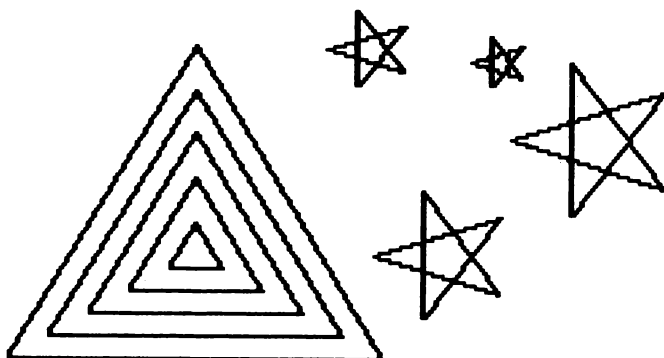



Figura 7.2 Stelle e triangoli in grande e in piccolo

SUGGERIMENTO

Chi impara il Logo dovrebbe ricordarsi come si usano le variabili (e le due regole presentate sopra, per evitare errori). Non è importante capire fin da ora la spiegazione del funzionamento delle variabili: con l'esperienza si riuscirà a capire meglio come mai le variabili funzionano così.
Ecco qui alcuni punti degni di nota per chi vuole approfondire l'argomento.
Il Logo si serve dei due punti e delle virgolette per segnalare fatti ben precisi:

1. :GRANDEZZA significa che il valore dato da tastiera va inserito nella casella chiamata "GRANDEZZA.
2. "GRANDEZZA significa il nome stesso, GRANDEZZA.
3. GRANDEZZA, senza due punti o virgolette, significa un comando o una procedura che ha nome GRANDEZZA.

Senza l'uso di questi due caratteri speciali, i due punti e le virgolette, vi sarebbe ambiguità in molte espressioni del linguaggio Logo. Per maggior chiarezza, conviene citare chiaramente questi segni anche quando si parla (magari dettando i comandi di una procedura a un amico). Ad esempio, per dire :GRANDEZZA, conviene citare a tutte lettere ... due punti GRANDEZZA, e per dire "GRANDEZZA ... virgolette GRANDEZZA.
C'è poi un altro fatto, che interessa quando si definisce una procedura, che prevede l'inserimento di un dato variabile, come ad esempio:

```
PER QUADRATO :GRANDEZZA
```

In questi casi il valore inserito nella casella "GRANDEZZA è valido solo all'interno della procedura. Se ad esempio si batte il comando QUADRATO 20, il valore 20 è posto nella casella "GRANDEZZA e vi rimane fino al termine della procedura QUADRATO, poi la casella viene vuotata.
Conviene provare un esempio:

```
STAMPA :GRANDEZZA
QUADRATO :GRANDEZZA
QUADRATO 20
STAMPA :GRANDEZZA
```

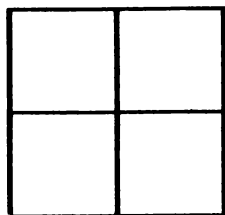
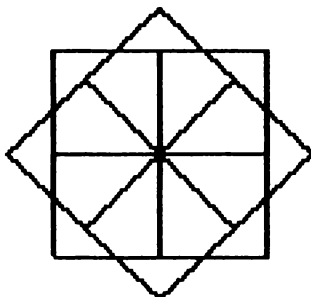
La casella "GRANDEZZA ha associato un valore solo mentre la procedura QUADRATO è attiva. Ecco un altro caso:

```
ASSEGNA "GRANDEZZA 50
STAMPA :GRANDEZZA
QUADRATO :GRANDEZZA
QUADRATO 20
STAMPA :GRANDEZZA
```

ASSEGNA è un modo alternativo di creare una variabile del Logo. Viene aperta una casella pubblica (cioè, non riservata a una particolare procedura) chiamata "GRANDEZZA nell'esempio, collocandovi il valore dato (nel nostro caso, 50). Nell'esempio considerato, vengono create così due caselle di nome "GRANDEZZA, una pubblica e l'altra privata, ancora vuota, e riservata alla procedura QUADRATO. Il comando STAMPA :GRANDEZZA si riferisce al valore della casella pubblica. Il comando QUADRATO :GRANDEZZA ricopia nella casella privata "GRANDEZZA il valore 50, che si trova nella casella pubblica con lo stesso nome. Quando poi si batte QUADRATO 20, il valore 50 viene sostituito dal valore 20 nella casella privata "GRANDEZZA (ma non in quella pubblica). Battendo ancora STAMPA :GRANDEZZA, si ottiene però 50, il valore contenuto nella casella "GRANDEZZA pubblica. È una faccenda un po' complicata, che sarà più chiara quando si vedrà per bene l'istruzione ASSEGNA.

7.2 Come cambiare la forma di una figura assegnando un valore

Gli esempi di Figura 7.3 sono realizzati con la stessa procedura, cambiando il valore di input, che rappresenta un angolo.
Ecco la procedura:

```
PER QUADRATOGIRA :ANGOLO
QUADRATO 50
DESTRA :ANGOLO
QUADRATOGIRA :ANGOLO
FINE
```

QUADRATOGIRA 90 QUADRATOGIRA 45 QUADRATOGIRA 15

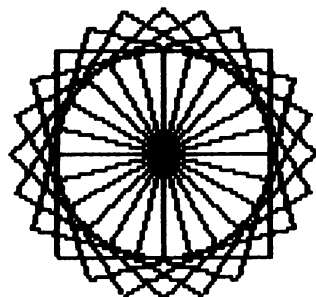



Figura 7.3 Immagini tracciate da QUADRATOGIRA con valori dell'angolo di 90, 45 e 15 gradi

La QUADRATOGIRA è una procedura recursiva, che cioè richiama se stessa (la tecnica recursiva è presentata nel paragrafo 5.4). Quando la QUADRATOGIRA esegue l'ultimo comando, prima di FINE, richiama una nuova procedura, identica a se stessa, chiamata anch'essa QUADRATOGIRA, e questa a sua volta ne richiama un'altra, e così via. La QUADRATOGIRA non si arresta da sola: bisogna fermarla battendo i tasti CTRL-G.
Il comando con cui una procedura richiama se stessa è detto comando recursivo.

TRABOCCHETTO

Quando si scrive una procedura recursiva che richiede l'assegnazione di dati di input, bisogna fare attenzione a ripetere la variabile di input anche nel comando recursivo. Come si vede nell'esempio di prima, la nuova QUADRATOGIRA deve avere assegnato il valore :ANGOLO, proprio come il comando DESTRA, nello stesso esempio.

Qui c'è un nuovo esempio, che traccia stelle di varia forma, assegnando diversi angoli (maggiori di 90 gradi):

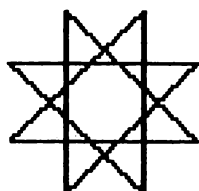
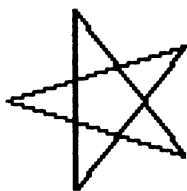
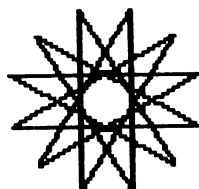
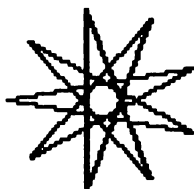
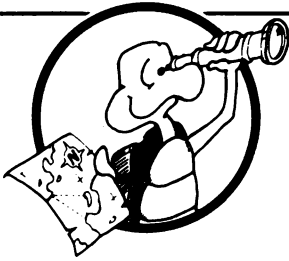
```
PER STELLE :ANGOLO
AVANTI 50
DESTRA :ANGOLO
STELLE :ANGOLO
FINE
```

STELLE 135

STELLE 144

STELLE 150

STELLE 160

Figura 7.4 Stelle tracciate dalla procedura STELLE, con angoli assegnati di 135, 144, 150 e 160 gradi

ESPLORAZIONE

- Tracciare tante stelle diverse, con la procedura STELLE :ANGOLO (basta assegnare ogni volta un valore diverso all'angolo). Come si fa a tracciare una stella a nove punte? E una a dieci punte? E una a quindici punte?
- Assegnare lo stesso numero, come input di QUADRATOGIRA e di STELLE. Provare varie volte, con numeri sempre nuovi e osservare il risultato (i due disegni hanno una certa somiglianza, perché vengono generati con lo stesso angolo).

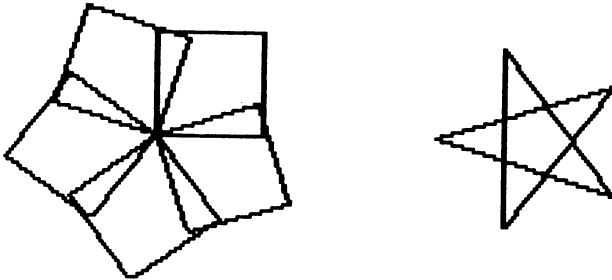



Figura 7.5 Disegni tracciati con QUADRATOGIRA 144 e STELLE 144

- Fare altri disegni con rotazione, seguendo l'ispirazione del momento.

7.3 Procedure con più di un dato di input

Ecco una procedura predisposta per tracciare stelle di angolo e grandezza variabile a piacere:

```
PER STELLE2 :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
STELLE2 :GRANDEZZA :ANGOLO
FINE
```

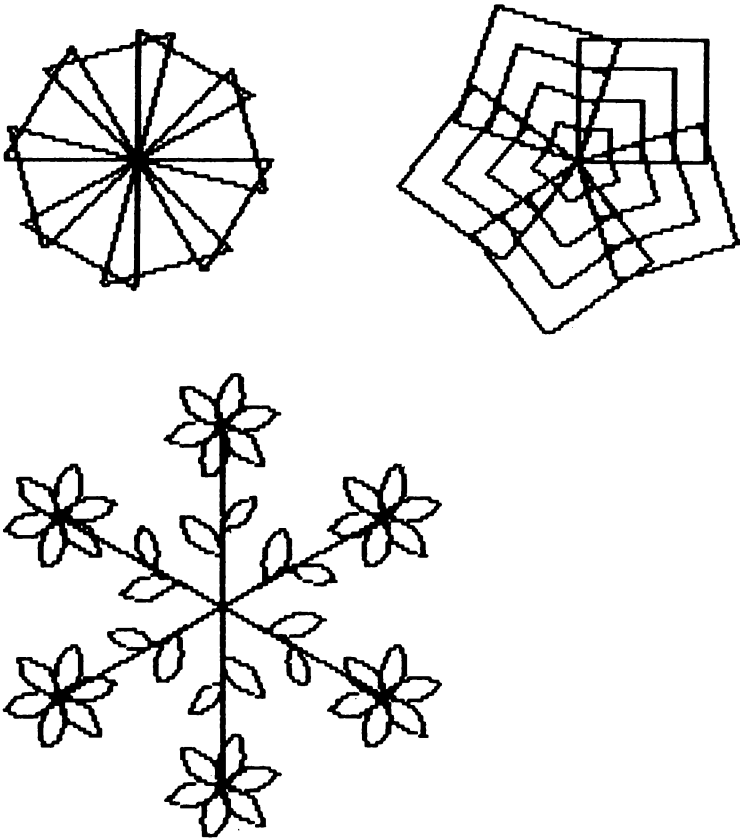



Figura 7.6 Disegni con rotazione, costruiti su forme elementari diverse

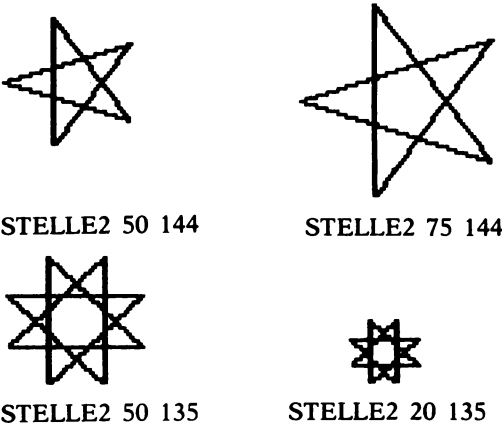



Figura 7.7 Disegni tracciati con STELLE2 :GRANDEZZA :ANGOLO

Le procedure di questo tipo sono spiegate nel Capitolo 8. Si può vedere qui ancora un esempio, che riguarda la costruzione di un rettangolo. Il rettangolo è definito da due numeri, che indicano la larghezza e l'altezza:

```
PER RETTANGOLO :ALTEZZA :LARGHEZZA
AVANTI :ALTEZZA
DESTRA 90
AVANTI :LARGHEZZA
DESTRA 90
AVANTI :ALTEZZA
DESTRA 90
AVANTI :LARGHEZZA
DESTRA 90
FINE
```

Con questa procedura, si può tracciare qualunque rettangolo, indicando la sua altezza e la sua larghezza.

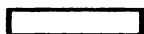
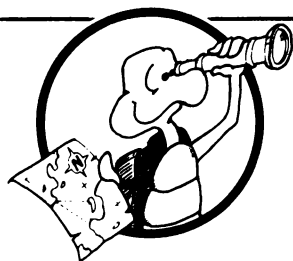



Figura 7.8 Rettangoli ottenuti con RETTANGOLO :ALTEZZA :LARGHEZZA

La procedura appena vista serve anche per tracciare quadrati (basta indicare altezza e larghezza uguali).

Vale la pena di provare a costruire disegni di fantasia, basati sulla rotazione di una figura elementare, come quelli di Figura 7.9.

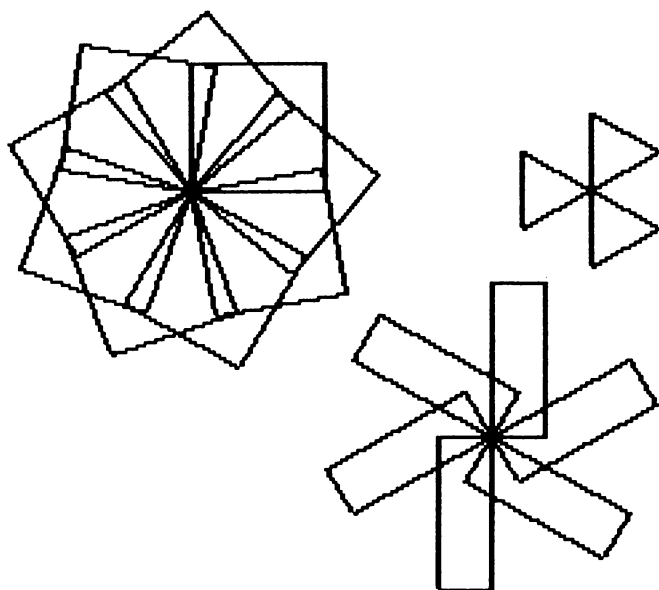



Figura 7.9 Disegni costruiti con la rotazione di quadrati, triangoli e rettangoli

7.4 Procedure secondarie con variabili

Anche le procedure secondarie (quelle che vengono richiamate all'interno di un'altra procedura) possono avere dati variabili. Ecco qui un esempio, ottenuto modificando la QUADRATOGIRA vista nel paragrafo 7.2 (nella prima versione, la QUADRATOGIRA richiama QUADRATO 50):

```
PER QUADRATOCHEGIRA :GRANDEZZA :ANGOLO
QUADRATO :GRANDEZZA
DESTRA :ANGOLO
QUADRATOCHEGIRA :GRANDEZZA :ANGOLO
FINE
```

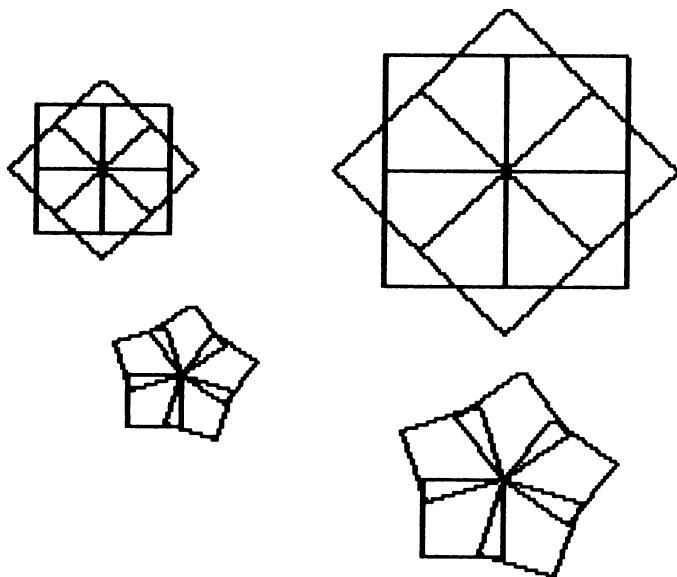



Figura 7.10 Ecco cosa può tracciare la QUADRATOCHEGIRA

Come si vede, invece di QUADRATO 50 si scrive QUADRATO :GRANDEZZA, e si aggiunge la variabile :GRANDEZZA, fra quelle che vanno assegnate quando si richiama la procedura QUADRATOCHEGIRA.
Continuiamo con altri esempi dell'uso delle variabili, supponendo di avere una procedura che traccia una casetta formata da un quadrato e da un triangolo, con 50 passi di lato, come in Figura 7.11. I comandi sono:

```
PER CASETTA
QUADRATO
AVANTI 50
DESTRA 30
TRIANGOLO
SINISTRA 30
INDIETRO 50
FINE
```

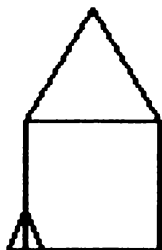



Figura 7.11 Ecco cosa traccia CASETTA

Si può ottenere una casetta di grandezza variabile a piacere, usando le procedure QUADRATO e TRIANGOLO con variabili e aggiungendo alla CASETTA una variabile di input:

```
PER CASETTA :GRANDEZZA
QUADRATO :GRANDEZZA
AVANTI :GRANDEZZA
DESTRA 30
TRIANGOLO :GRANDEZZA
SINISTRA 30
INDIETRO :GRANDEZZA
FINE
```

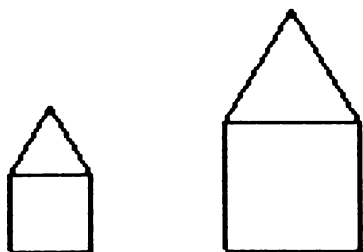



Figura 7.12 Ecco cosa tracciano CASETTA 20 e CASETTA 50

Qui c'è un altro esempio, che parte da una procedura che traccia una fila di casette, con questi comandi:

```
PER CASETTE
PARTAVAN (-100)
CASETTA 10
PARTAVAN 10
CASETTA 20
```

```
PARTAVAN 20
CASETTA 30
PARTAVAN 30
CASETTA 40
PARTAVAN 40
FINE
```

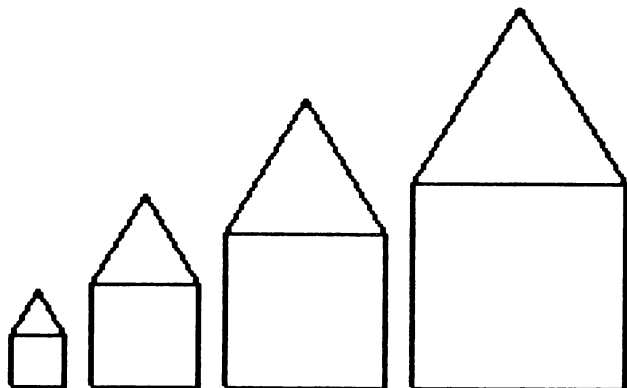
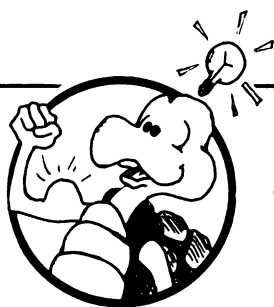



Figura 7.13 Fila di case tracciate con CASETTE

Come si vede, viene usata la procedura PARTAVAN, vista già nel paragrafo 6.1 (dove si è costruito il camion). Qui PARTAVAN ha assegnato, la prima volta, un valore negativo, per far muovere la tartaruga verso sinistra, invece che verso destra.

Quando si assegna a una procedura un valore negativo, conviene sempre scriverlo fra parentesi, come in (−100). Se si scrive −100, senza parentesi, il Logo può fraintendere, e credere di dover eseguire una sottrazione (da un numero che invece non esiste).

Ecco qui la PARTAVAN usata in CASETTE (è simile alla PARTAVAN già vista nel paragrafo 6.1, ma con la possibilità di assegnare un valore a piacere):

```
PER PARTAVAN :GRANDEZZA
SULAPENNA
DESTRA 90
AVANTI :GRANDEZZA+10
SINISTRA 90
GIULAPENNA
FINE
```

IDEA

Avete visto cosa può fare il Logo? Tratta le variabili come tratta ogni altro numero, sommandole, sottraendole, moltiplicandole, ecc. Nell'esempio, si vede una somma (si trova nell'istruzione AVANTI :GRANDEZZA+10).
Ecco una proprietà davvero interessante, che verrà molto usata più avanti.
Attenzione, quando si batte il segno + (più) o il segno − (meno), a non tenere premuto il tasto SHIFT, perché il Logo protesta (anche se sullo schermo compare il segno "+") e compare il messaggio:

LA PROCEDURA + NON ESISTE

Ricordate le procedure FIORE e COROLLA, del Capitolo 6? Si possono trasformare facilmente in procedure con variabili, come si è fatto con le altre:

```
PER PETALO :GRANDEZZA
ARCODES :GRANDEZZA
DESTRA 90
ARCODES :GRANDEZZA
DESTRA 90
FINE
```

```
PER COROLLA :GRANDEZZA
RIPETI 6[PETALO :GRANDEZZA DESTRA 60]
FINE

PER STELO :GRANDEZZA
AVANTI :GRANDEZZA
PETALO :GRANDEZZA
AVANTI :GRANDEZZA*2
FINE

PER FIORE :GRANDEZZA
STELO :GRANDEZZA
COROLLA :GRANDEZZA
INDIETRO :GRANDEZZA*3
FINE
```

Come si vede, in FIORE e in STELO compare una moltiplicazione. Come mai la tartaruga deve avanzare di :GRANDEZZA*2 passi in STELO e indietreggiare di :GRANDEZZA*3 passi in FIORE?
Per chi ancora non lo sa, il simbolo "*" indica al Logo una moltiplicazione.
È facile, ora, fare un giardinetto:

```
PER GIARDINETTO
PARTAVAN (-50)
FIORE 30
PARTAVAN 20
FIORE 40
PARTAVAN 20
FIORE 50
FINE
```

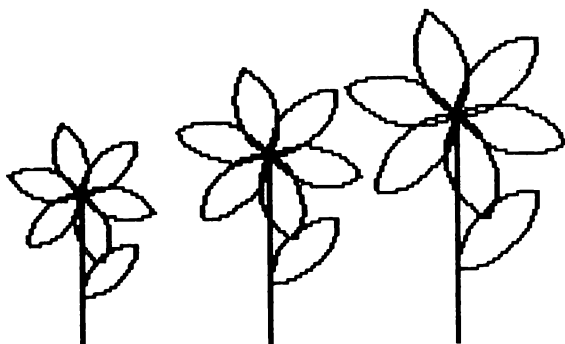
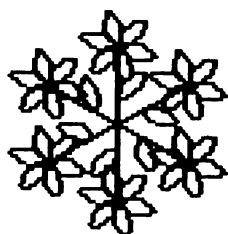



Figura 7.14 Giardinetto con fiori di varia grandezza

INFIORESCENZA 20 60 INFIORESCENZA 40 45

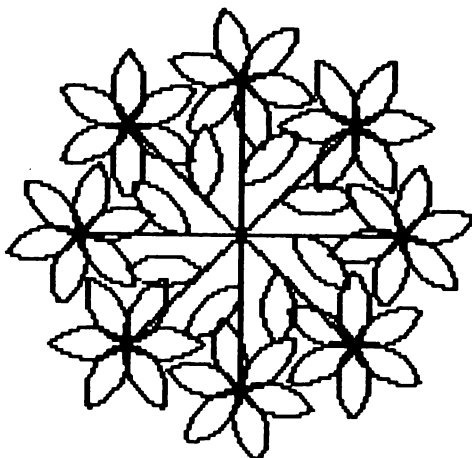
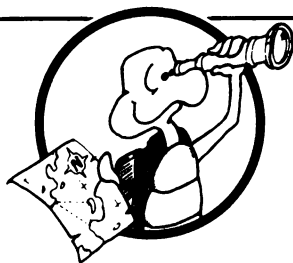



Figura 7.15 Disegni fatti con INFIORESCENZA :GRANDEZZA :ANGOLO

Con FIORE si possono tracciare anche disegni più elaborati, come quello in Figura 7.15:

```
PER INFIORESCENZA :GRANDEZZA :ANGOLO
FIORE :GRANDEZZA
DESTRA :ANGOLO
INFIORESCENZA :GRANDEZZA :ANGOLO
FINE
```

ESPLORAZIONE

- Si possono ora riprendere i disegni fatti nei Capitoli 5 e 6, inserendovi le variabili. Con le nuove procedure, si possono preparare disegni nuovi e più interessanti (è facile ingrandire o ridurre una parte, in modo che si adatti bene all'insieme). La Figura 7.16 mostra alcuni esempi.
- Con un po' di fantasia, si possono inventare altre procedure che impiegano variabili in input.

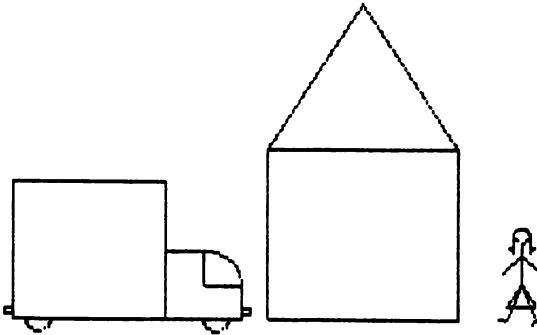



Figura 7.16 Disegni già realizzati nei Capitoli 5 e 6, con la modifica della grandezza delle parti

7.5 Disegni che crescono e che si arrestano da soli

Abbiamo già visto alcuni casi di disegni che crescono, come quello tracciato dalla RIQUADRO nel paragrafo 7.1:

```
PER RIQUADRO
QUADRATO 10
QUADRATO 20
QUADRATO 30
QUADRATO 40
QUADRATO 50
FINE
```

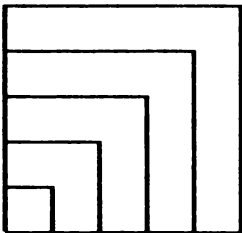



Figura 7.17 Disegno tracciato con RIQUADRO

Anche la procedura CASETTE (vista nel paragrafo 7.4) traccia oggetti che si ingrandiscono man mano:

```
PER CASETTE
PARTAVAN (-100)
CASETTA 10
PARTAVAN 10
CASETTA 20
PARTAVAN 20
CASETTA 30
PARTAVAN 30
CASETTA 40
PARTAVAN 40
FINE
```

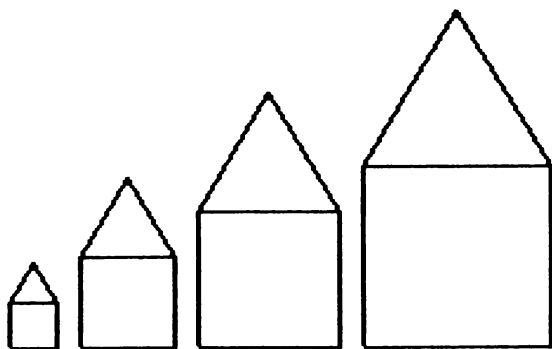
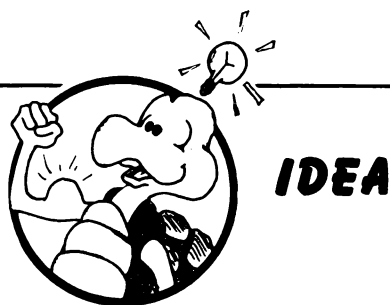



Figura 7.18 Ecco una casetta che cresce, cresce, cresce ...

È facile modificare queste procedure, inserendo delle variabili, per avere un ingrandimento progressivo:

```
PER QUADROCRESCE :GRANDEZZA
QUADRATO :GRANDEZZA
QUADROCRESCE :GRANDEZZA+10
FINE

PER CASACRESCE :GRANDEZZA
CASETTA :GRANDEZZA
PARTAVAN :GRANDEZZA
CASACRESCE :GRANDEZZA+10
FINE
```

Il valore assegnato all'inizio, quando si richiama QUADROCRESCE o CASACRESCE, indica la grandezza di partenza. Provate con CASACRESCE 10 e con QUADROCRESCE 10, ad esempio; per fermare il disegno, bisogna battere la coppia di tasti CTRL-G.

Le procedure QUADROCRESCE e CASACRESCE non fanno mica molto: in fondo, tutto il disegno è tracciato dalle procedure secondarie CASETTA, QUADRATO e PARTAVAN. La funzione di QUADROCRESCE e di CASACRESCE è solo quella di modificare il proprio valore di input, che indica la grandezza del nuovo elemento da tracciare! Conviene osservare bene la QUADROCRESCE, comando per comando. Supponiamo di battere:

QUADROCRESCE 10
: QUADROCRESCE parte con il valore 10 nella sua casella di input, e richiama la QUADRATO.

QUADRATO :GRANDEZZA
: QUADRATO riceve nella sua casella di input il valore 10 passato da QUADROCRESCE, e traccia il quadrato di lato 10. Al termine la QUADROCRESCE richiama se stessa, aggiungendo 10 al valore di input.

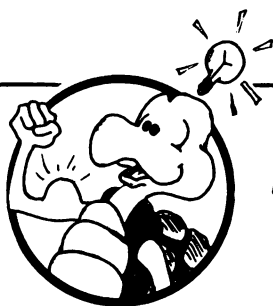
QUADROCRESCE :GRANDEZZA+10
: Con questo comando la QUADROCRESCE aggiunge 10 al valore della sua casella di input (che si chiama "GRANDEZZA) e assegna il valore 20 così ottenuto alla casella di input di una nuova procedura QUADROCRESCE e si ricomincia da capo.

QUADRATO :GRANDEZZA
: Questa volta la QUADRATO è richiamata con il valore 20.

QUADROCRESCE :GRANDEZZA+10
: Viene richiamata una nuova QUADROCRESCE, con il valore 30, cioè 20+10 e così via...

È chiaro che QUADROCRESCE non si ferma mai. Se si prova a battere il comando:

QUADROCRESCE 10

si vede cosa succede (per fermarsi, si batte CTRL-G).

Eppure ci dev'essere un modo per far fermare la procedura da sola! Ecco cosa si fa: si indica alla procedura di arrestarsi quando il valore di :GRANDEZZA supera un limite assegnato.
Ed ecco come si fa: supponiamo, per esempio, che il limite da noi scelto per la grandezza sia 100. Si aggiunge allora alla procedura QUADROCRESCE questo comando:

```
SE :GRANDEZZA >100 ALLORA STOP
```

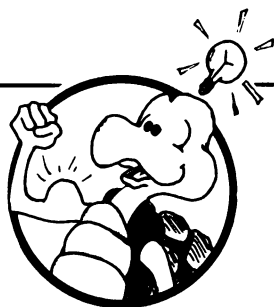
Qui vediamo due comandi nuovi, SE e STOP (ALLORA fa parte del comando SE e serve solo per rendere più chiara la scrittura del comando: se si vuole, si può anche fare a meno di scriverlo, perché è sottinteso).
Dove va inserito il nuovo comando? Bastano pochi tentativi per scoprirlo da soli. Anzi, proviamo senz'altro tutte le possibilità (che sono solo tre):

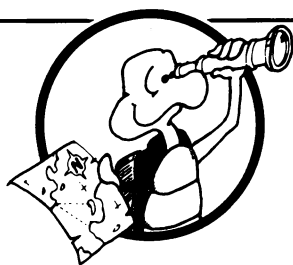
```
PER QUADROCRESCE1 :GRANDEZZA
QUADRATO :GRANDEZZA
QUADROCRESCE1 :GRANDEZZA+10
SE :GRANDEZZA >100 ALLORA STOP
FINE

PER QUADROCRESCE2 :GRANDEZZA
QUADRATO :GRANDEZZA
SE :GRANDEZZA >100 ALLORA STOP
QUADROCRESCE2 :GRANDEZZA+10
FINE

PER QUADROCRESCE3 :GRANDEZZA
SE :GRANDEZZA >100 ALLORA STOP
QUADRATO :GRANDEZZA
QUADROCRESCE3 :GRANDEZZA+10
FINE
```

In uno dei tre casi, la procedura non si ferma, in un altro si ferma dopo

aver tracciato il quadrato di lato 100 e nell'altro ancora si arresta dopo aver tracciato il quadrato di lato 110.
Si riesce a dire quale caso non si ferma mai? E quale si arresta al quadrato di lato 100? La verifica è facile: basta provare.

TRABOCCHETTO

In QUADROCRESCE1 c'è un errore! Veramente, sembra proprio naturale aggiungere il nuovo comando (qualcuno dice istruzione, invece di comando, perché, in effetti, si tratta di più comandi scritti in una riga sola) alla fine, dopo quelli già esistenti. Invece, se ci si pensa appena un poco, si capisce che così il comando non funziona, perché il computer non lo esegue mai: infatti, quando viene eseguito il comando della seconda riga, il Logo richiama una nuova procedura QUADROCRESCE1, che ricomincia da capo.
Anche in QUADROCRESCE2 c'è un piccolo bug, di tipo diverso. La procedura si arresta, però solo dopo aver tracciato un quadrato con il lato maggiore del limite indicato, 100. Il controllo per lo STOP è qui inserito dopo il comando QUADRATO, e per questo QUADROCRESCE2 prima traccia un quadrato e poi controlla se è troppo grande.
QUADROCRESCE3 invece controlla la grandezza del nuovo quadrato prima di tracciarlo, e quando :GRANDEZZA vale 110 si arresta in tempo.

IDEA

QUADROCRESCE è un esempio di una tecnica fondamentale di programmazione: mentre si esegue una serie di elaborazioni con l'uso di variabili, si controlla che una condizione di stop sia rispettata e se lo è si arresta l'elaborazione; se invece la condizione non è rispettata, si prosegue con i nuovi valori delle variabili appena considerati.

ESPLORAZIONE

- Aggiungere un controllo di stop a CASACRESCE e provare se funziona proprio come si vuole.
- Costruire altri disegni che si ingrandiscono progressivamente, basati su forme elementari diverse e provare ad inserire anche qui il controllo di stop.

SUGGERIMENTO

Il modo migliore per far capire questo tipo di procedimento è seguire riga per riga una procedura, come si è fatto con QUADROCRESCE. Questo modo di fare, in cui ci si mette in un certo senso nei panni del computer (proprio come prima ci si è messi nei panni di Turtle) è molto utile e si può spingere fino al livello della singola procedura, che può essere considerata come un "omino" che deve eseguire un compito ben preciso (volendo, si può anche organizzare una specie di rappresentazione, con diversi attori che impersonano ciascuno una procedura diversa).
È importante, se si rappresenta una procedura recursiva come QUADROCRESCE, che ogni volta sia un attore nuovo a presentarsi in scena (non bisogna pensare che sia la stessa procedura che "ricomincia da capo", perché si tratta ogni volta di una procedura nuova). Nel caso di QUADROCRESCE, l'attore che la impersona deve chiamare un nuovo attore e assegnargli un nuovo valore di input (nell'esempio visto, è il vecchio valore di input aumentato di dieci). Il primo attore non esce di scena, ma aspetta che il nuovo attore abbia fatto la sua parte, per dire la battuta (il comando) seguente del suo copione (nel nostro esempio, è semplicemente FINE).
Attenzione a questo particolare, perché non è facile afferrare l'idea che una procedura rimanga in attesa fino a che la procedura da essa richiamata non sia terminata.
Questa spiegazione può sembrare eccessiva, perché la QUADROCRESCE, scelta come esempio, è piuttosto semplice. L'approccio presentato si rivela però molto utile nel caso di procedure più complesse, dove veramente conviene "rappresentare" mentalmente ciascuna procedura, come se recitasse una parte. In fondo, l'esempio semplice vale quanto quello complicato, se lo si intende nel modo giusto; d'altronde, ogni caso concreto fa storia a sé, e gli esempi non sono fatti per risolvere in anticipo tutti i problemi (cosa impossibile) ma per dare la possibilità di capirli e quindi di risolverli, quando si presentano. Non dimentichiamo che l'esempio è una metafora.

7.6 Altre procedure che crescono e si arrestano da sole

Possiamo riprendere ora la procedura QUADRATOCHEGIRA, vista nel paragrafo 7.4:

```
PER QUADRATOCHEGIRA :GRANDEZZA :ANGOLO
QUADRATO :GRANDEZZA
DESTRA :ANGOLO
QUADRATOCHEGIRA :GRANDEZZA :ANGOLO
FINE
```

Come si vede, la procedura ha due variabili di input: si può provare con diversi valori di grandezza e di angolo, per tracciare forme come quelle di Figura 7.19.
Aggiungiamo ora un controllo di stop alla QUADRATOCHEGIRA, per farla arrestare dopo che ha completato il disegno per la prima volta.

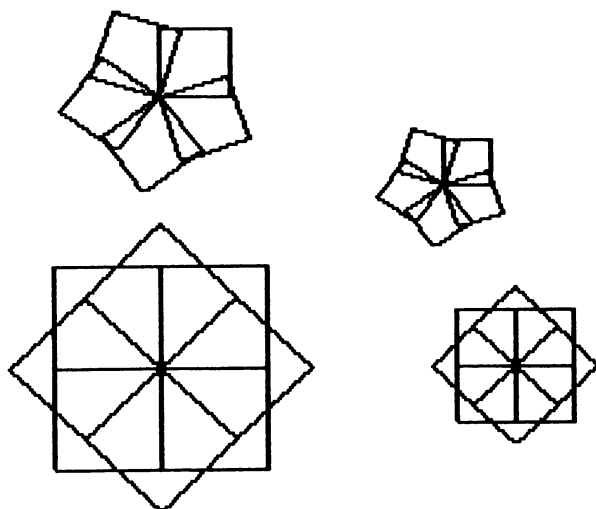
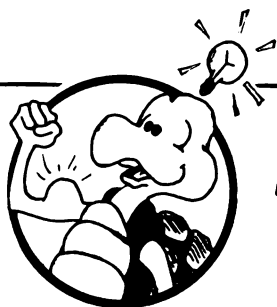



Figura 7.19 Forme tracciate con QUADRATOCHEGIRA

Qui non si sa quanti quadrati formano il disegno, perché questo dipende dall'angolo assegnato in input. Il modo più semplice di controllare lo stop è di arrestare la procedura quando la tartaruga ritorna alla posizione di partenza, con lo stesso orientamento (che in questo caso è verticale, rivolto verso l'alto, cioè con DIREZIONE uguale a zero).
Ecco la modifica:

```
PER QUADROGIRAEFERMA :GRANDEZZA :ANGOLO
QUADRATO :GRANDEZZA
DESTRA :ANGOLO
SE DIREZIONE = 0 ALLORA STOP
QUADROGIRAEFERMA :GRANDEZZA :ANGOLO
FINE
```

QUADROGIRAEFERMA si arresta dopo aver tracciato il disegno completo una volta sola.
La nuova istruzione funziona così: DIREZIONE è un comando del Logo che fa sapere qual è la direzione verso cui è rivolta la tartaruga (in gradi, rispetto alla verticale, in senso orario). Si può ad esempio provare questo caso:

```
DISEGNA
DESTRA 45
STAMPA DIREZIONE
45
DESTRA 90
STAMPA DIREZIONE
135
ecc.
```

Anche il segno = è un comando del Logo, che richiede due numeri da paragonare: il risultato è la parola "VERO se i due numeri sono uguali, la parola "FALSO se sono diversi. Si può provare l'esempio che segue:

STAMPA 5 = 4+1
VERO
STAMPA 3 = 0
FALSO
ecc.

SE è un altro comando del Logo, che ammette in input solo due parole, "VERO e "FALSO. Se la parola assegnata è "VERO, vengono eseguiti i comandi scritti nel seguito della riga (in questo caso, il comando STOP), mentre se la parola è "FALSO il computer passa senz'altro ad eseguire l'istruzione scritta nella riga seguente.
ALLORA non è un comando vero e proprio, ma serve solo a rendere più chiara la scrittura del comando SE.

SUGGERIMENTO

Vediamo bene questa istruzione:

SE DIREZIONE = 0 ALLORA STOP

Qui si trovano ben quattro comandi originali del Logo (chiamati anche primitive, in gergo). Ogni comando può essere pensato come un omino che deve eseguire un compito ben preciso: questi omini hanno il compito di scambiarsi messaggi l'uno con l'altro.
Per prima cosa, SE cerca il proprio valore di input e lo richiede a =, fornendogli i due input che questo si aspetta: il primo è il numero 0, il secondo è l'indicazione del comando DIREZIONE, che fornirà il numero richiesto.
A questo punto, = paragona il valore fornito da DIREZIONE con il numero 0: se i due numeri fossero uguali, invierebbe a SE la parola "VERO, ma non lo sono in questo caso, per cui invia la parola "FALSO.
Ora, finalmente, SE ha un valore su cui lavorare. Se il valore fosse "VERO, allora farebbe proseguire la QUADROGIRAEFERMA per il resto dell'istruzione, con il comando STOP che arresta la procedura. In questo caso, invece, il valore è "FALSO, e allora fa saltare la procedura all'istruzione successiva.
Il procedimento continua: di nuovo l'omino SE fa la sua comparsa, e chiama l'omino =. Quando la tartaruga punta in alto, l'omino DIREZIONE risponde all'omino = con il valore 0.

In questo caso, l'omino = risponde a SE con la parola "VERO, e questo fa quindi eseguire il resto dell'istruzione, che è il comando STOP, facendo arrestare la QUADROGIRAEFERMA che al momento è attiva; ora può riprendere la procedura che aveva chiamato la QUADROGIRAEFERMA e che stava in attesa (nel nostro caso, è un'altra QUADROGIRAEFERMA), e così via. In pratica, c'è una fila di procedure QUADROGIRAEFERMA in attesa di proseguire, e tutte hanno come istruzione successiva il comando FINE.

Proviamo ora con una nuova procedura, basata sulla QUADROGIRAEFERMA e con l'accrescimento automatico. Bisogna trovare un nome nuovo, ad esempio QUAGIFERCRE:

```
PER QUAGIFERCRE :GRANDEZZA :ANGOLO
SE :GRANDEZZA >100 STOP
QUADROGIRAEFERMA :GRANDEZZA :ANGOLO
QUAGIFERCRE (:GRANDEZZA+20) :ANGOLO
FINE
```

Le parentesi nella penultima istruzione non sono richieste dal Logo, ma servono per rendere più chiara la scrittura dell'istruzione, indicando che (:GRANDEZZA+20) è un singolo valore di input per QUAGIFERCRE.
Il valore dell'angolo determina la forma del disegno che si ottiene, mentre la grandezza dipende dal secondo valore dato alla procedura. In Figura 7.20 si vedono due esempi.

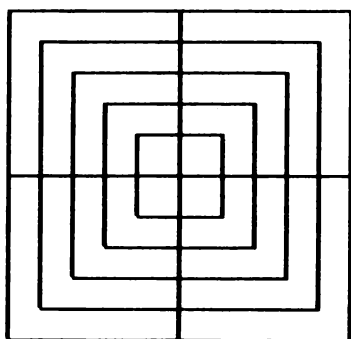
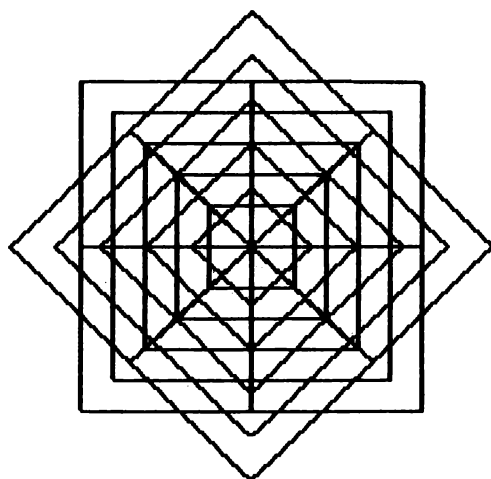
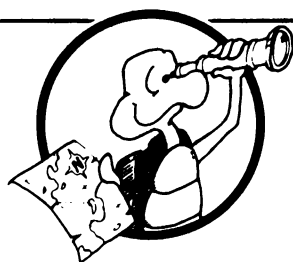



Figura 7.20 Forme tracciate con QUAGIFERCRE :GRANDEZZA :ANGOLO

ESPLORAZIONE

- Provare la QUAGIFERCRE più volte, assegnando angoli diversi (il valore assegnato come grandezza non importa molto).
- Eliminare il controllo di stop e vedere che cosa succede, quando il disegno si avvolge intorno allo schermo.
- Preparare altre procedure come la QUAGIFERCRE, basate su forme elementari diverse dal quadrato, come il triangolo, o magari un fiore...

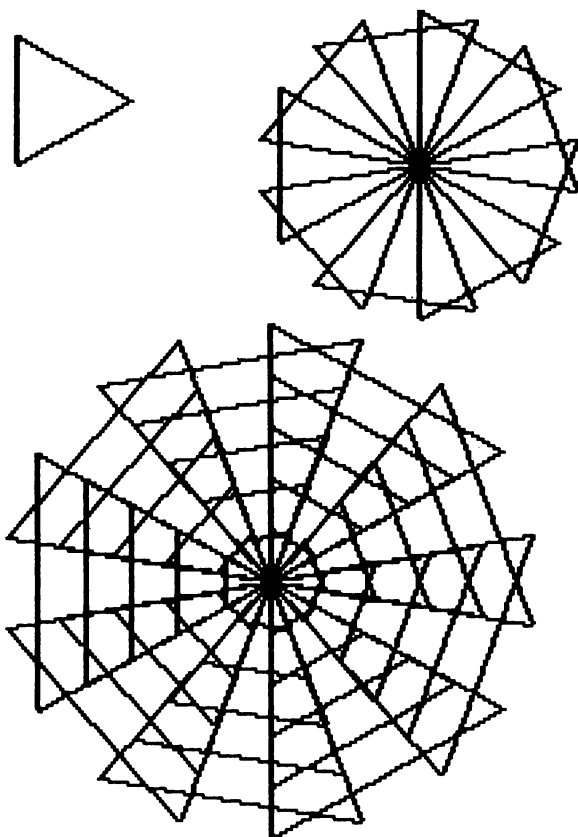



Figura 7.21 Ragnatele fatte con triangoli

8

POLIGONALE e i suoi simili

In questo capitolo si vedono nuovi modi di creare disegni geometrici e di esplorarne l'effetto, a volte sorprendente. Chi apprezza il lato matematico dei disegni geometrici trova qui l'occasione di nuove interessanti scoperte, e chi preferisce disegnare e basta può avere una gran quantità di spunti e di suggerimenti pratici sul disegno con la tartaruga. Se si vuole invece lasciar perdere il disegno, almeno per il momento, si può passare direttamente al Capitolo 9, che riguarda il dialogo con il computer.

Comando	**Abbreviazione**	**Esempi**
ASSEGNA		ASSEGNA "PARTENZA DIREZIONE ASSEGNA "GRANDEZZA 50
TRACCIA NOTRACCIA		
—		STAMPA 35 – 10 AVANTI :GRANDEZZA – 10
*		STAMPA 3*5, AVANTI :GRANDEZZA*3

Non vengono usati file dal disco di servizio
Non vengono usate nuove procedure di servizio

8.1 POLIGONALE

Nel paragrafo 7.3 abbiamo visto una procedura chiamata STELLE2, che traccia stelle di varia forma e grandezza: in realtà la STELLE2 può fare molto di più, perché riesce a tracciare anche poligoni. La riprendiamo qui, con un nuovo nome, POLIGONALE, per sviluppare tutte le sue possibilità:

```
PER POLIGONALE :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
POLIGONALE :GRANDEZZA :ANGOLO
FINE
```

Come si vede, POLIGONALE richiede due valori, una lunghezza e un angolo: la prima istruzione della procedura fa tracciare alla tartaruga una riga della lunghezza indicata, la seconda istruzione ruota la tartaruga dell'angolo indicato; la terza istruzione richiama una nuova POLIGONALE che si comporta come la prima (ma ora la posizione e la direzione della tartaruga sono cambiate...), e così via. Da sola, la POLIGONALE non si ferma (bisogna battere CTRL-G). Ecco qui, in Figura 8.1, che cosa si riesce a fare con POLIGONALE.

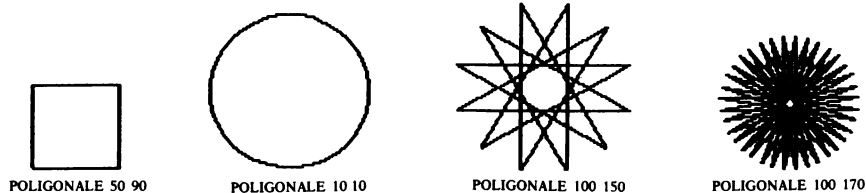
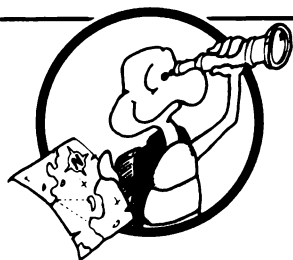



Figura 8.1 Disegni tracciati con POLIGONALE :GRANDEZZA :ANGOLO

POLIGONALE può fare molte cose; mentre la si esplora, conviene fare un po' d'ordine sin dall'inizio, scrivendo quello che si fa nel diario del Logo. Si prende nota dei dati assegnati in input e si scrive accanto che cosa

succede. I disegni più interessanti possono essere contrassegnati, in modo da ritrovarli subito, quando li si vuole riprendere. In pratica, va bene un prospetto come quello riportato qui sotto:

Grandezza (passi)	*Angolo (gradi)*	*Poligono (P) o Stella (S)*	*Numero di lati*	*Si avvolge attorno allo schermo*
30	30	P	12	no
60	30	P	12	sì
30	60	P	6	no
30	80	S	9	no
ecc.				

Ecco qualche spunto per esplorare cosa può fare POLIGONALE:

- Provare numeri scelti a caso, per la grandezza e l'angolo. Vanno bene numeri anche molto piccoli o molto grandi. Certi effetti sono sorprendenti! Provare ad esempio con POLIGONALE 99 99, oppure con POLIGONALE 12 345, o POLIGONALE 6 600, ecc.
- Prendere un numero per la grandezza, ad esempio 50, e tenerlo fisso, mentre si cambia ogni volta il valore dell'angolo, come in POLIGONALE 50 75, POLIGONALE 50 100, POLIGONALE 50 150, ecc.
- Tenere fissa la grandezza e variare l'angolo, in modo da ottenere i seguenti poligoni: quadrato, triangolo, esagono (ha sei lati), ottagono (ha otto lati) e ennagono (ha nove lati). Più difficile è tracciare un pentagono (ha cinque lati). L'ettagono (con sette lati) è ancora più impegnativo.
- Tenere ancora fissa la grandezza, e cercare di ottenere (cambiando l'angolo) stelle diverse: a cinque punte, otto punte, nove punte, ecc. Scrivere i tentativi e i risultati sul diario, in ordine.
- Tenere fisso, ora, l'angolo e provare a cambiare la grandezza, come in POLIGONALE 10 100, POLIGONALE 50 100, POLIGONALE 100 100, ecc.
- Assegnare valori molto piccoli per la grandezza e l'angolo, come in POLIGONALE 1 2, POLIGONALE 2 5, POLIGONALE 5 1, ecc.
- Assegnare una grandezza intermedia, come 50 o 100, e angoli piccolissimi, come 1 o 2:

```
POLIGONALE 100 1
POLIGONALE 100 2
POLIGONALE 100 3
ecc.
```

- Assegnare una grandezza intermedia con angoli molto grandi, come 1000 o 20000:

POLIGONALE 100 1000
POLIGONALE 100 5000
POLIGONALE 100 20000
ecc.

- Usare angoli "normali", come 60 o 150 e grandezze grandissime o piccolissime:

    POLIGONALE 1 150
    POLIGONALE 3 60

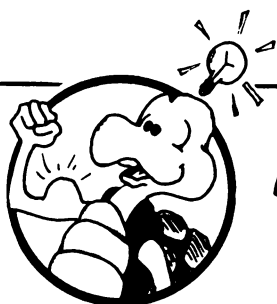
  oppure

    POLIGONALE 1000 150
    POLIGONALE 5000 60
    ecc.

- Provare con valori uguali per grandezza e angolo, piccoli, medi, grandi e grandissimi:

    POLIGONALE 1 1
    POLIGONALE 2 2
    POLIGONALE 20 20
    POLIGONALE 90 90
    POLIGONALE 150 150
    POLIGONALE 1000 1000
    ecc.

- Provare alcuni angoli molto particolari:

    POLIGONALE 20 0
    POLIGONALE 20 180
    POLIGONALE 20 360
    ecc.

In un poligono, il numero di lati ha una relazione matematica ben precisa con l'angolo assegnato (nella stella, vale la stessa relazione per il numero di punte). Si riesce a scoprire qual è? Conoscendo questa regola, è facile prevedere la forma che si ottiene con POLIGONALE; anzi, ci si può costruire un piccolo gioco, in cui bisogna indovinare il numero di lati (o di punte) conoscendo l'angolo assegnato (la prova si fa direttamente col computer). Ecco qui alcune domande (il primo gruppo di angoli produce poligoni, il secondo stelle):

- A quanti lati di poligono corrispondono gli angoli di 30, 60, 90, 120 e 180 gradi?
- Si riesce a scoprire la regola?
- Quante punte ha la stella con angolo 80, 150, 160 e 200 gradi?
- Scoprire anche qui la regola matematica.

La regola da scoprire ha a che fare con il numero 360: in effetti, ogni volta che ritorna al punto di partenza, e con lo stesso orientamento, la tartaruga fa anche almeno un giro completo su se stessa, di 360 gradi, appunto; può anche fare più giri completi. Questa è la cosiddetta regola del percorso chiuso: un percorso chiuso è fatto da una serie di passi che riportano la tartaruga al punto di partenza, con lo stesso orientamento iniziale.
Detta in altro modo, la regola stabilisce che la tartaruga, seguendo un percorso chiuso, ruota di 360 gradi, o di un multiplo intero di 360 gradi (ad esempio può ruotare due volte su se stessa, per 720 gradi, o tre volte, per 1080 gradi, ecc.).
A questo punto, forza e coraggio! La regola è a portata di mano: conoscendola, si potrà usare la POLIGONALE a colpo sicuro.

8.2 Come fermare la POLIGONALE

Sarebbe bello poter fermare la POLIGONALE quando ha completato il disegno la prima volta invece di lasciarla continuare a ripetere quello che ha fatto. Basta inserire un comando di stop su condizione, facendo controllare ad ogni passo al computer se la tartaruga ha raggiunto di nuovo la posizione e l'orientamento di partenza; ecco come si fa:

```
PER POLIGONALE :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SE DIREZIONE = 0 STOP
POLIGONALE :GRANDEZZA :ANGOLO
FINE
```

Con il comando DIREZIONE, si ottiene l'orientamento della tartaruga; quando parte dopo un comando DISEGNA, la tartaruga è sempre orientata in direzione 0 gradi. Come si vede, qui si controlla se la direzione è 0 e se lo è si arresta la procedura, perché vuol dire che la tartaruga è di nuovo al punto di partenza (in POLIGONALE, posizione e orientamento sono legati fra loro). Provare è facile: se tutto funziona a dovere, si riesce a costruire disegni come quello di Figura 8.2.

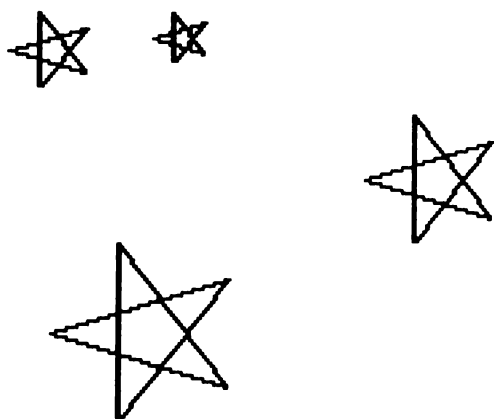



Figura 8.2 Disegno fatto con POLIGONALE

MIGLIORIAMO IL CONTROLLO DI STOP

Se la POLIGONALE parte con la tartaruga orientata non in su, a 0 gradi, ma in un'altra direzione, ci sono problemi di funzionamento: a volte la procedura si arresta prima di aver completato il disegno, in altri casi non si arresta per nulla. Bisogna modificare allora la procedura per renderla più generale. Con il comando EDITA, si interviene allora così:

```
PER POLIGONALE :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SE DIREZIONE = :PARTENZA STOP
POLIGONALE :GRANDEZZA :ANGOLO
FINE
```

Si usa, come si vede, una nuova variabile chiamata :PARTENZA, per indicare l'orientamento iniziale della tartaruga. Non abbiamo però ancora indicato al computer il valore di :PARTENZA. Se si batte il comando:

```
POLIGONALE 50 100
```

il Logo protesta:

```
MANCA IL NOME PARTENZA ...
```

Un modo per assegnare il valore di :PARTENZA fa intervenire il comando ASSEGNA:

```
ASSEGNA "PARTENZA DIREZIONE
```

ASSEGNA richiede due valori: il primo è il nome di una variabile, il secondo è il valore da assegnare a questa. Il comando ASSEGNA "PARTENZA DIREZIONE assegna alla variabile "PARTENZA il valore fornito da DIREZIONE (che è l'orientamento della tartaruga, nel momento in cui viene eseguito il comando). Battendo ora:

```
POLIGONALE 50 100
```

si ottiene un altro effetto, perché :PARTENZA ha un valore. Proviamo allora la POLIGONALE, dopo aver ripulito lo schermo e ruotato la tartaruga in una qualunque direzione:

```
DISEGNA
DESTRA 99
```

ASSEGNA "PARTENZA DIREZIONE
POLIGONALE 45 45

(qualunque angolo va bene, da assegnare a DESTRA)

TRABOCCHETTO

Attenzione ai simboli ! Il primo valore da fornire a ASSEGNA è sempre un nome, che nel Logo si distingue perché incomincia con il segno ", come ad esempio "GIOVANNI, "LAURA o "PARTENZA. Il simbolo : è usato invece per indicare il valore di una variabile. Allora, "PARTENZA non è lo stesso di :PARTENZA.
Per provare, basta battere:

STAMPA "PARTENZA
STAMPA :PARTENZA

A questo punto, si può migliorare la POLIGONALE dal punto di vista del suo uso pratico: sarebbe noioso dover battere ogni volta l'istruzione ASSEGNA "PARTENZA DIREZIONE, così si prepara una nuova superprocedura che lavora per noi: la chiameremo, per brevità, POLGON1. Ecco qui:

PER POLGON1 :GRANDEZZA :ANGOLO
ASSEGNA "PARTENZA DIREZIONE
POLIGONALE :GRANDEZZA :ANGOLO
FINE

E ora proviamo:

POLGON1 50 90
DESTRA 45
POLGON1 50 90

Come si vede, non importa come parte la tartaruga, perché questa può avere qualunque posizione e qualunque orientamento. Se tutto funziona a

puntino, POLGON1 deve tracciare il suo disegno e fermarsi da sola. In Figura 8.3 ci sono alcuni disegni fatti con POLGON1: vale la pena di provarli, e magari di colorarli, se si ha un monitor TV a colori.

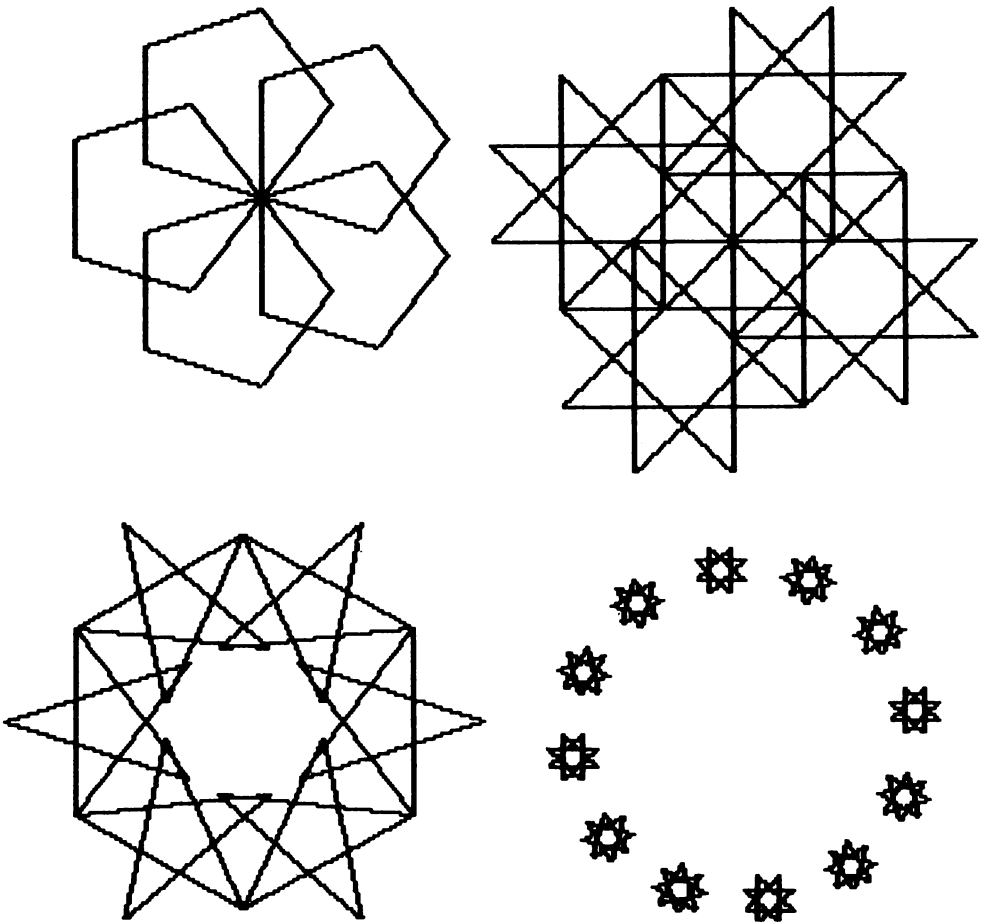
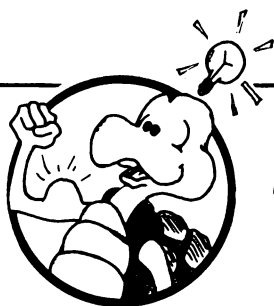
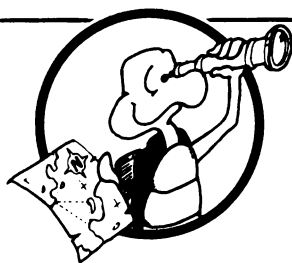



Figura 8.3 POLGON1 può far questo e altro

POLGON1 è una superprocedura un po' particolare. Il suo unico compito è preparare la partenza di un'altra procedura, che fa tutto il lavoro. In questo caso, assegna il valore alla variabile "PARTENZA, poi richiama POLIGONALE che usa il valore :PARTENZA per arrestarsi. Questo modo di fare è utile in molte occasioni, specie quando si lavora con programmi (è un altro modo di chiamare le procedure) complicati, e verrà seguito spesso, d'ora in poi.

8.3 Tutto sui controlli di stop

Il lettore interessato ai controlli di stop trova qui suggerimenti ed esempi utili in molti casi pratici. Chi invece preferisce proseguire con nuovi tipi di disegno può passare senz'altro al paragrafo 8.4 che presenta le spirali segmentate e ritornare qui in un secondo momento, quando trova un problema di stop che non riesce a risolvere bene.

L'aspetto forse più importante di un controllo di stop è la sua posizione nella procedura. Qui sotto, ad esempio, ci sono quattro procedure uguali in tutto, fuorché nel nome e nella posizione del controllo di stop. Provandole tutte, ci si rende conto della differenza:

```
PER POLGONA :GRANDEZZA :ANGOLO
SE DIREZIONE = 0 STOP
AVANTI :GRANDEZZA
DESTRA :ANGOLO
POLGONA :GRANDEZZA :ANGOLO
FINE

PER POLGONB :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
SE DIREZIONE = 0 STOP
DESTRA :ANGOLO
POLGONB :GRANDEZZA :ANGOLO
FINE

PER POLGONC :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SE DIREZIONE = 0 STOP
POLGONC :GRANDEZZA :ANGOLO
FINE

PER POLGOND :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
POLGOND :GRANDEZZA :ANGOLO
SE DIREZIONE = 0 STOP
FINE
```

Uno dei bug più comuni nei programmi per computer è proprio la posizione sbagliata di un comando su condizione, come il controllo di stop appena visto. Vale la pena di capire bene la differenza fra le quattro procedure presentate sopra, perché così si è in grado di individuare bene questo tipo di bug. Come al solito, bisogna seguire ogni procedura passo passo, e si deve arrivare a vedere chiaramente come e perché la POLGONC funziona bene, e le altre tre no.

SUGGERIMENTO

L'esplorazione sui controlli di stop può essere molto utile per chi impara, specie se si riesce a inculcare i due concetti presentati qui di seguito.

1. Le procedure vanno seguite passo passo, mettendosi sempre nei panni del computer che le esegue, proprio come si fa quando si segue il movimento della tartaruga. Dopo questa interpretazione mentale, si prova la procedura per vedere che cosa succede realmente.
   Il Logo dispone del comando TRACCIA, che è fatto apposta per stanare i bug. Se si batte il comando TRACCIA, il Logo esegue le procedure un passo alla volta, e non passa al successivo se non si batte il tasto RETURN.
   Se si batte:

   ```
   TRACCIA
   DISEGNA
   POLGONA 100 90
   ```

   compare sullo schermo la prima istruzione:

   ```
   SE DIREZIONE = 0 STOP
   ```

   Non succede niente altro finché non si batte il tasto RETURN. Ecco, qui si deve pensare: "Qual è la direzione della tartaruga?" La direzione è zero, perché la tartaruga non si è ancora mossa; allora la procedura si deve fermare. Battendo il tasto RETURN si vede che cosa succede alla POLGONA; compare la scritta:

   ```
   POLGONA SI FERMA
   ```

   Ora si ripulisce lo schermo battendo il comando DISEGNA e si riprova con POLGONB, POLGONC e POLGOND.
   Per annullare TRACCIA, si batte il comando NOTRACCIA.
2. Bisogna sempre cercare di spiegare perché ci aspettiamo che il computer si comporti in un certo modo, poi provare che cosa fa realmente, e spiegare come mai si comporta proprio così. Con i computer, si fa spesso molta confusione perché ci si aspetta che accadano cose diverse da quelle che poi realmente accadono. Questa difficoltà vale per tutti, ed è importante rendersene conto e superarla nel modo appena descritto.
   La spiegazione del comportamento del computer può essere fatta in molti modi, e spesso uno vale l'altro. È importante però fare in modo che ognuno si senta libero di dare la propria spiegazione, a suo modo, senza

provare imbarazzo o sentirsi esposto al giudizio degli altri, perché l'apprendimento dipende proprio dal confronto fra ciò che ci si aspetta e ciò che accade realmente, e dall'analisi della differenza
In una classe, ci può essere talvolta qualche difficoltà a stabilire questo clima rilassato, perché gli allievi possono essere abituati a parlare solo quando conoscono la risposta "giusta": questa abitudine va contrastata, perché in questo campo valgono anche le risposte "sbagliate", se le si sa valutare per quello che sono. Al contrario, bisogna fare di tutto per stabilire un clima di libero scambio di opinioni, su come si crede che funzioni il computer, perché così si riesce veramente a cogliere le idee sbagliate e a correggerle.
Per un genitore, o un insegnante, il modo migliore di "rompere il ghiaccio" è di parlare dei propri errori passati, in fatto di computer, e di come si è riusciti a correggerli: questo, più di ogni altra cosa, renderà gli allievi fiduciosi di essere compresi se parleranno liberamente delle loro idee sull'argomento.

8.4 Spirali segmentate

Nel Capitolo 7 abbiamo visto come si impiegano le variabili per costruire figure che si accrescono da sole, come in QUADROCRESCE e in QUAGIFERCRE.

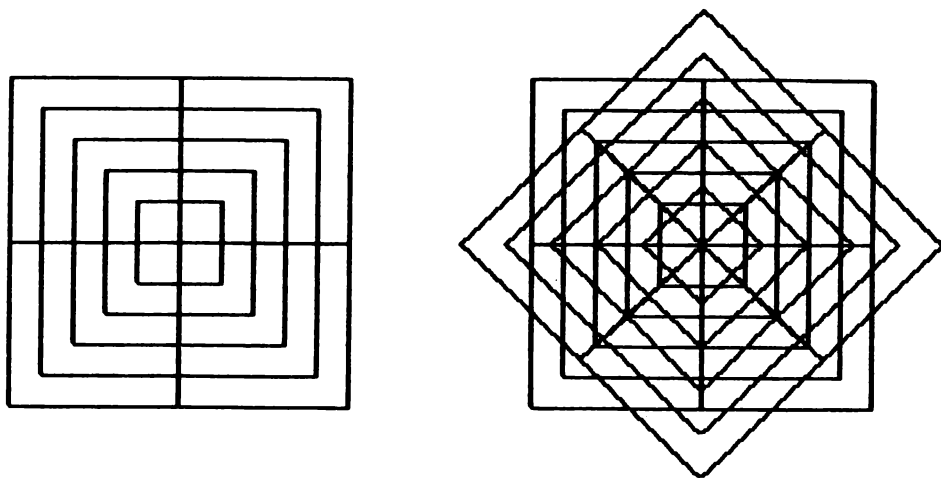



Figura 8.4 QUADROCRESCE e QUAGIFERCRE possono fare anche questo

Le spirali segmentate sono simili ai poligoni.
Esse sono figure che si accrescono. In pratica, ad ogni passo della spirale segmentata, il valore della :GRANDEZZA viene aumentato, oppure diminuito (ci sono due tipi di spirali).
Ecco qui una spirale che si espande ad ogni passo:

```
PER SPIRASEG :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SPIRASEG (:GRANDEZZA+1) :ANGOLO
FINE
```

In Figura 8.5 compaiono due esempi di spirali generate da SPIRASEG, con diversi valori di :ANGOLO

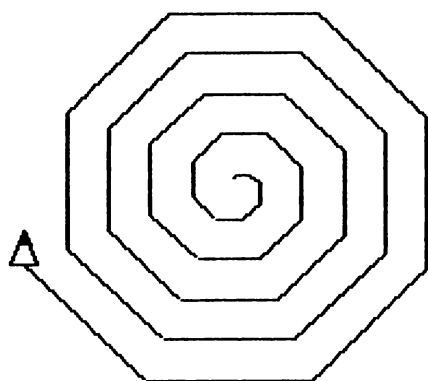
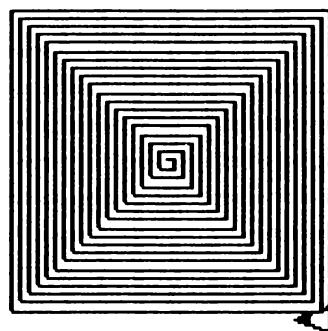
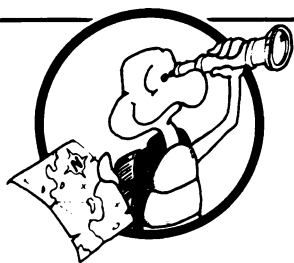
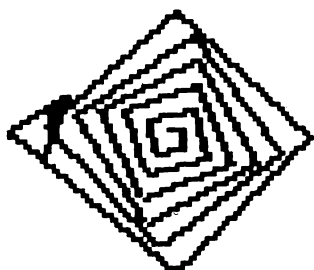
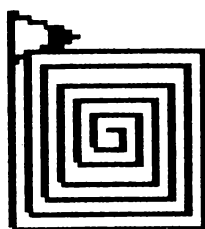
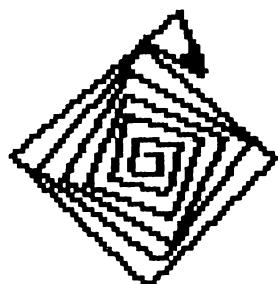
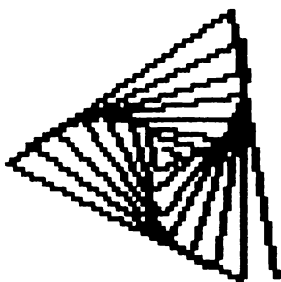
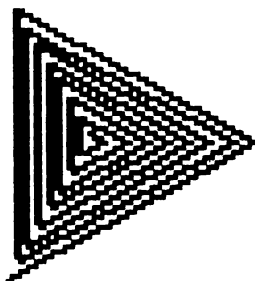



Figura 8.5 Ecco due tipi di spirale segmentata

ESPLORAZIONE

- Provare SPIRASEG molte volte, con angoli sempre diversi. Si ottengono effetti interessanti con valori prossimi (ma non uguali) a quelli che generano una stella, o un poligono.

SPIRASEG 1 88 | SPIRASEG 1 90 | SPIRASEG 1 92

SPIRASEG 1 118 | SPIRASEG 1 120 | SPIRASEG 1 122

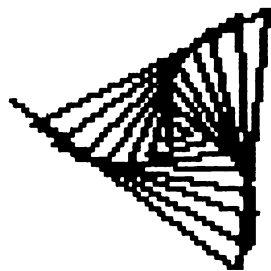



Figura 8.6 Spirali segmentate a volontà

- Lasciar crescere qualche spirale finché diventa molto grande, prima di fermarla con CTRL-G, e vedere che effetto si ottiene quando il disegno si avvolge attorno allo schermo.

Se si modifica l'incremento assegnato ad ogni passo alla grandezza del segmento, si ottengono effetti interessanti. Si aggiunge in pratica, un'altra variabile:

```
PER SPIRASEGVAR :GRANDEZZA :ANGOLO :AUMEN
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SPIRASEGVAR (:GRANDEZZA+:AUMEN) :ANGOLO :AUMEN
FINE
```

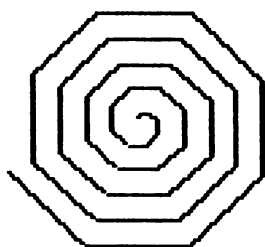
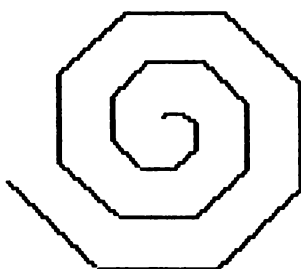



Figura 8.7 Spirali con diverso incremento

La funzione di :AUMEN è chiara, in quanto aumenta ogni volta la lunghezza del segmento che viene tracciato.

- Provare SPIRASEGVAR molte volte, cambiando il valore di :AUMEN. Si può anche completare la procedura con un controllo di stop, come si vede qui:

```
PER SPISEGVARSTOP :GRANDEZZA :ANGOLO :AUMEN
SE :GRANDEZZA>100 STOP
AVANTI :GRANDEZZA
```

```
DESTRA :ANGOLO
SPISEGVARSTOP (:GRANDEZZA+:AUMEN) :ANGOLO :AUMEN
FINE
```

Con le spirali c'è un'altra possibilità di costruzione: si incomincia con un segmento lungo e lo si riduce ad ogni passo (è il contrario di quello che si è fatto finora).
Ecco un caso:

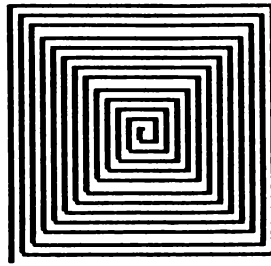
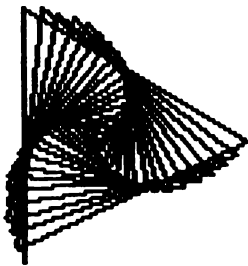
```
PER SPIRIDUSTOP :GRANDEZZA :ANGOLO :RIDUZ
SE :GRANDEZZA<1 STOP
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SPIRIDUSTOP (:GRANDEZZA- :RIDUZ) :ANGOLO :RIDUZ
FINE
```

SPIRIDUSTOP 100 122 2   SPIRIDUSTOP 100 90 2

Figura 8.8 Spirali con segmenti che si accorciano

Con le spirali ci si può davvero sbizzarrire. Nella Figura 8.9 ci sono alcuni esempi, costruiti con più spirali (il segreto sta nella scelta di valori dell'angolo di rotazione che si adattino bene l'uno con l'altro).
Un'altra variazione interessante è tracciare solo gli angoli, e non i lati della spirale (vedi Figura 8.10). Con l'aggiunta dei comandi SULAPENNA e GIULAPENNA il gioco è fatto:

```
PER SPIRANGO :GRANDEZZA :ANGOLO :RIDUZ
SE :GRANDEZZA<1 STOP
SULAPENNA AVANTI :GRANDEZZA
GIULAPENNA AVANTI 1 INDIETRO 1
DESTRA :ANGOLO
SPIRANGO (:GRANDEZZA- :RIDUZ) :ANGOLO :RIDUZ
FINE
```

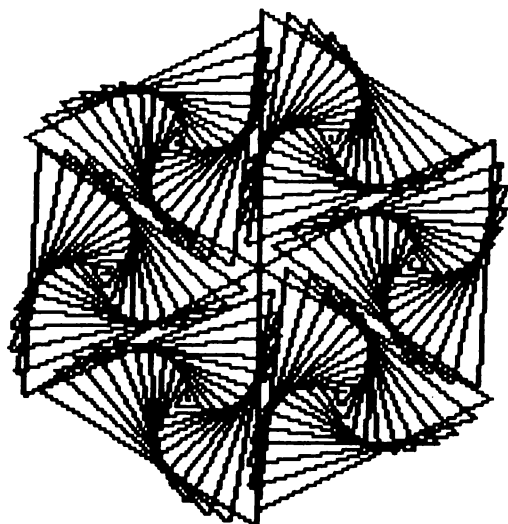
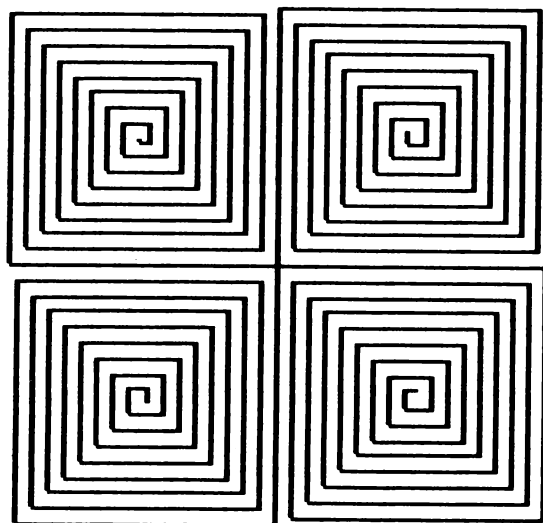



Figura 8.9 Combinazioni di spirali che si riducono su se stesse

In Figura 8.11 si vede come cambia l'effetto se si varia l'angolo, con incrementi di 5 gradi da un caso all'altro.

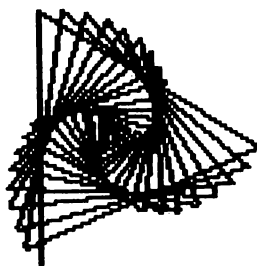



Figura 8.10 Spirali visualizzate per intero e solo per angoli

8.5 Volute

Se ad ogni passo si modifica non la lunghezza del segmento, ma l'ampiezza dell'angolo, si traccia una voluta invece di una spirale:

```
PER VOLUTA :GRANDEZZA ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
VOLUTA :GRANDEZZA (:ANGOLO+10)
FINE
```

L'incremento dell'angolo può essere assegnato ad una variabile; si regola così facilmente anche la forma della voluta, come si vede provando questa modifica:

```
PER VOLUTA2 :GRANDEZZA :ANGOLO :AUMEN
AVANTI :GRANDEZZA
DESTRA :ANGOLO
VOLUTA2 :GRANDEZZA (:ANGOLO+:AUMEN) :AUMEN
FINE
```

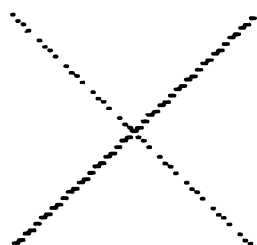
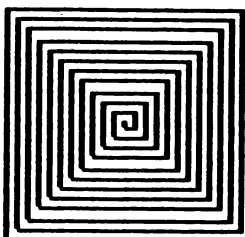
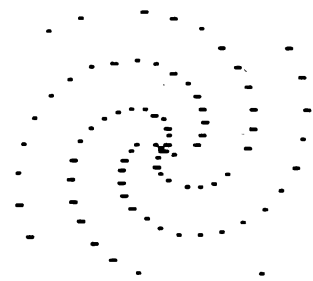
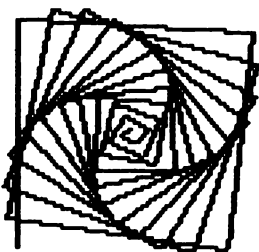
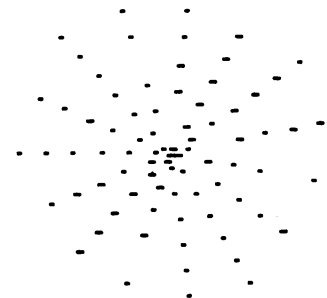
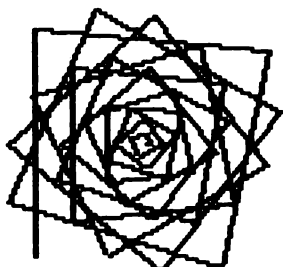
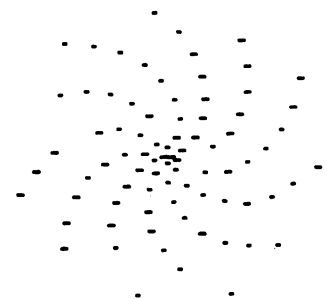
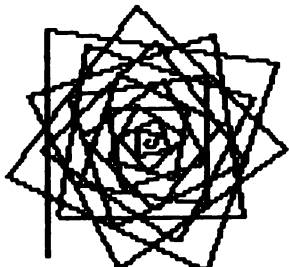



Figura 8.11 Effetto di piccoli cambiamenti dell'angolo della spirale

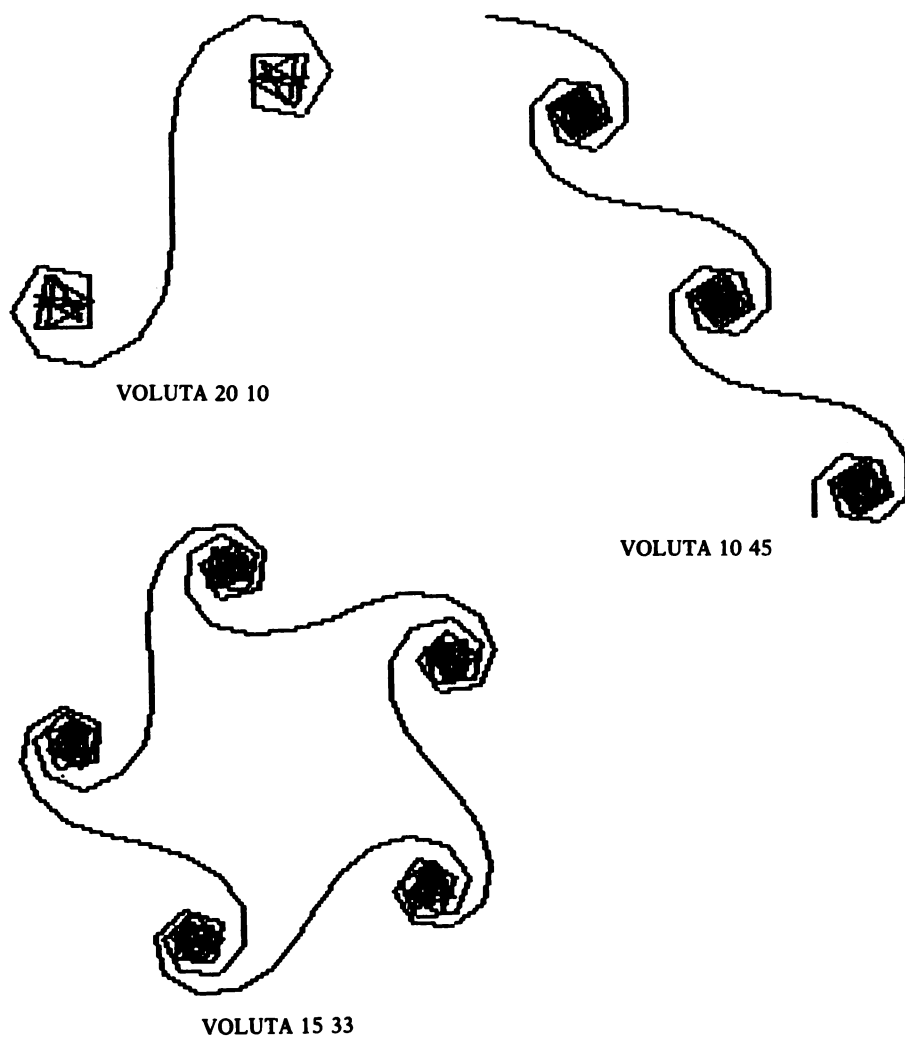



Figura 8.12 Volute

Se si varia anche l'orientamento di partenza della tartaruga, oltre che l'incremento dell'angolo di rotazione, si ottengono effetti veramente sorprendenti.

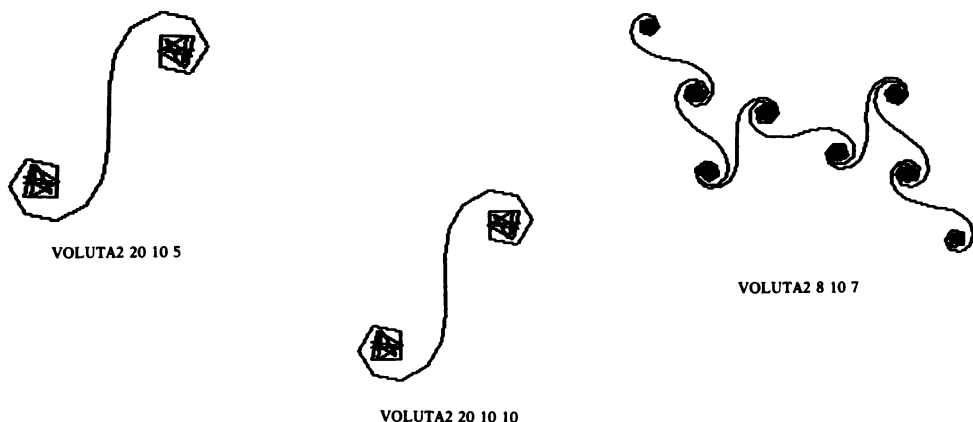
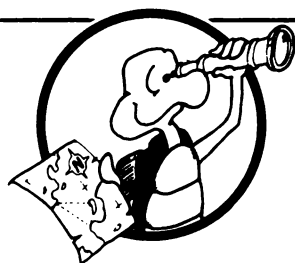



Figura 8.13 Volute realizzate con diversi incrementi angolari

Sono molti i disegni che si possono tracciare cambiando il secondo e il terzo argomento (i valori assegnati alle procedure si chiamano anche così) della VOLUTA2. La teoria matematica delle volute è molto complessa: qui ci sono, comunque, alcuni punti che vale la pena approfondire:

- Quali combinazioni di argomenti della VOLUTA2 producono disegni con solo due nodi?
- Quali combinazioni producono disegni che non ripassano mai dal punto di partenza?

Conviene scrivere sul diario le proprietà delle volute che man mano si scoprono, provando e riprovando.

8.6 Altre variazioni su POLIGONALE

Le variazioni sul tema della POLIGONALE sono veramente tantissime, e molto interessanti: ci sono addirittura libri dedicati interamente ai disegni geometrici che si possono fare con la tartaruga.

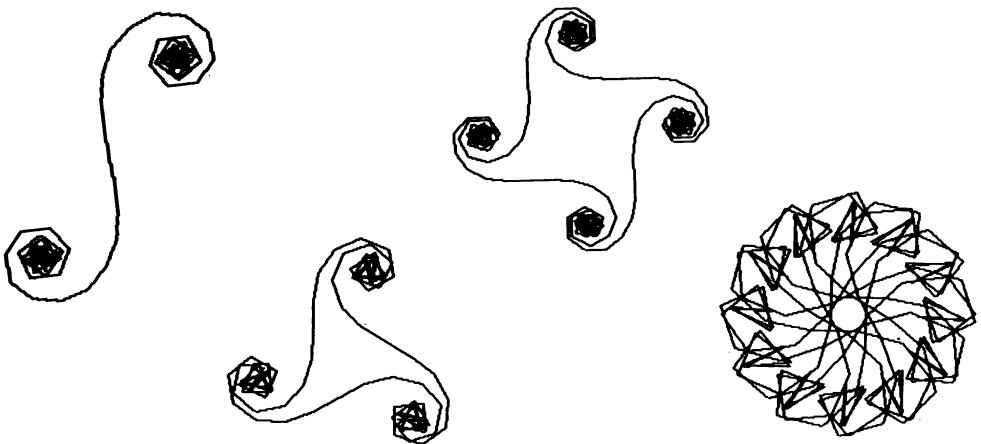



Figura 8.14 Volute a due, tre, quattro e dodici nodi, prodotte con VOLUTA2

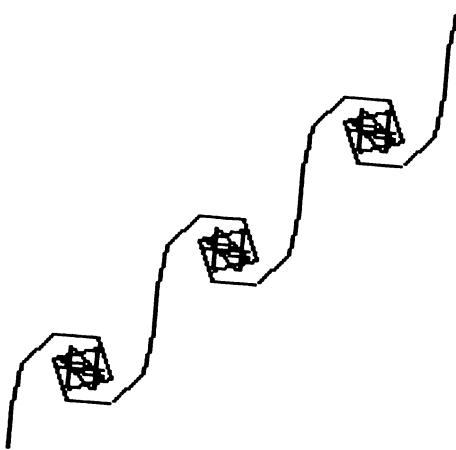



Figura 8.15 Voluta che non ritorna mai su se stessa

Ecco la prima variazione sul tema:

```
PER POLDOPPIO :G1 :A1 :G2 :A2
AVANTI :G1 DESTRA :A1
AVANTI :G2 DESTRA :A2
POLDOPPIO :G1 :A1 :G2 :A2
FINE
```

Questa procedura usa due passi di incremento invece di uno solo, e ri-

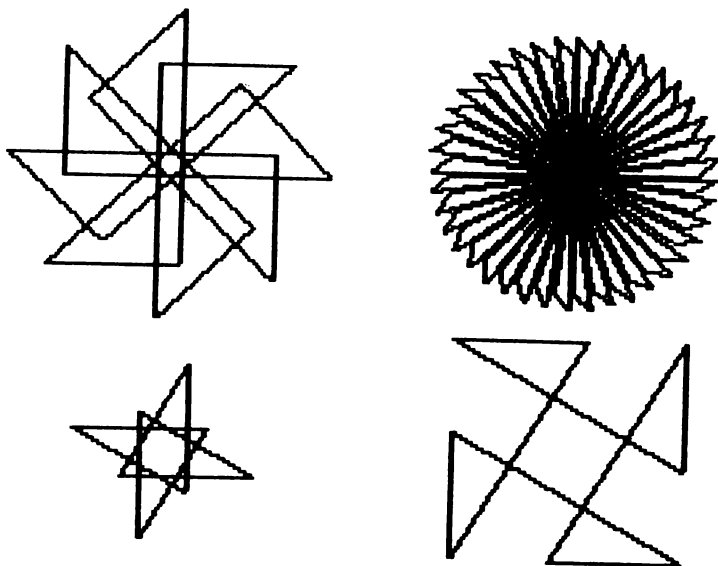



Figura 8.16 Disegni fatti con POLDOPPIO

chiede i valori :G1 e :G2, che stanno per :GRANDEZZA1 e :GRANDEZZA2, e i valori :A1 e :A2, che stanno per :ANGOLO1 e :ANGOLO2 (in totale, sono quattro valori). È interessante provare diverse combinazioni di valori, anche negativi (i valori negativi vanno racchiusi fra parentesi, così il Logo non cerca di fare una sottrazione, che qui non può essere fatta, ma accetta il numero negativo così com'è). Ecco un caso da provare:

```
POLDOPPIO 50 45 (-25) 90
```

L'avanzamento negativo vale come il ritorno indietro. Dire alla tartaruga AVANTI (−25) è come dire INDIETRO 25, e dire DESTRA (−30) vale come SINISTRA 30. Provare per credere.

Un'altra variazione, piuttosto semplice, di POLIGONALE, si fa scambiando ad ogni passo l'incremento di angolo e quello di grandezza, come si vede qui:

```
POLALTERNO :GRANDEZZA :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SE DIREZIONE = 0 STOP
POLALTERNO :ANGOLO :GRANDEZZA
FINE
```

Come si vede, nell'ultima istruzione prima di FINE, :GRANDEZZA e :ANGOLO si sono scambiate di posto. La POLALTERNO è davvero molto divertente, se la si prova un po'.
Ormai la strada è aperta: si può andare avanti con ogni genere di modifica. Ad esempio, invece di fare solo un tratto avanti e una rotazione ad

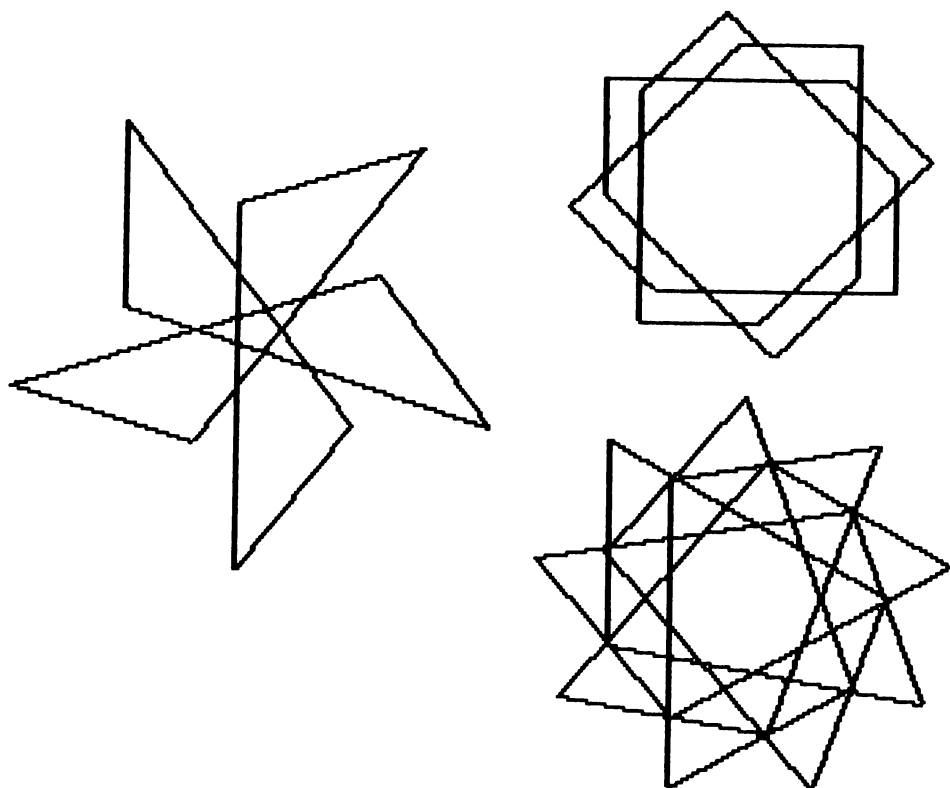



Figura 8.17 Disegni fatti con **POLALTERNO**

ogni passo, si può anche disegnare una figura, magari un triangolo, come qui:

```
PER POLITRANG :GRANDEZZA :ANGOLO
TRIANGOLO :GRANDEZZA
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SE DIREZIONE = 0 STOP
POLITRANG :GRANDEZZA :ANGOLO
FINE
```

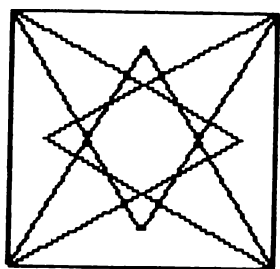
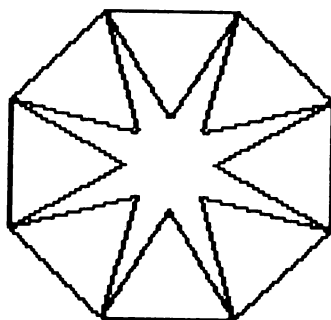
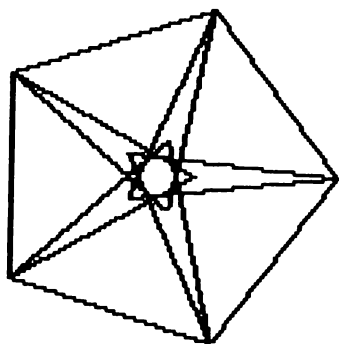



Figura 8.18 POLITRANG in azione

Naturalmente va prima definita la procedura TRIANGOLO :GRANDEZZA. Al posto del triangolo, si capisce, va bene qualunque altro motivo, come lo zig zag di Figura 8.19, che si traccia con questa procedura, molto semplice:

```
PER ZIGZAG :GRANDEZZA :ANGOLO
DESTRA :ANGOLO
AVANTI :GRANDEZZA
```

```
SINISTRA 2* :ANGOLO
AVANTI :GRANDEZZA
DESTRA :ANGOLO
FINE
```

ZIGZAG 40 60

Figura 8.19 Motivo a zig zag

Inserendo il motivo a zig zag in POLIGONALE, si ottengono effetti molto belli:

```
PER POLIZAG :GRANDEZZA :ANGOLO
ZIGZAG :GRANDEZZA :ANGOLO
DESTRA :ANGOLO
POLIZAG :GRANDEZZA :ANGOLO
FINE
```

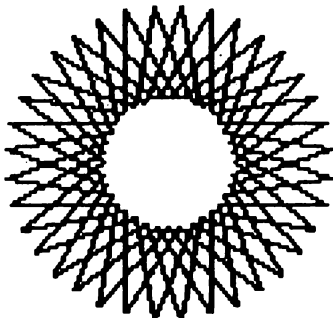
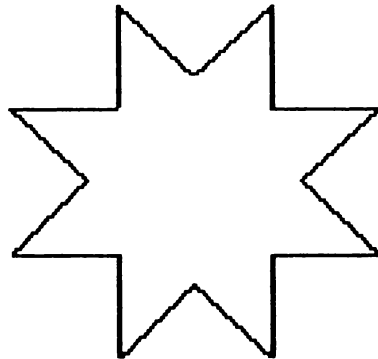
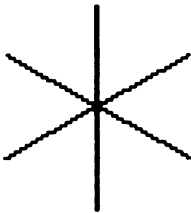



Figura 8.20 POLIZAG!

Se si assegnano due angoli diversi, per la rotazione di POLIGONALE e il motivo a zig zag, le sorprese non finiscono più, superando veramente l'immaginazione:

```
PER POLANZAGAN: GRANDEZZA :A1 :A2
ZIGZAG :GRANDEZZA :A1
DESTRA :A2
POLANZAGAN :GRANDEZZA :A1 :A2
FINE
```

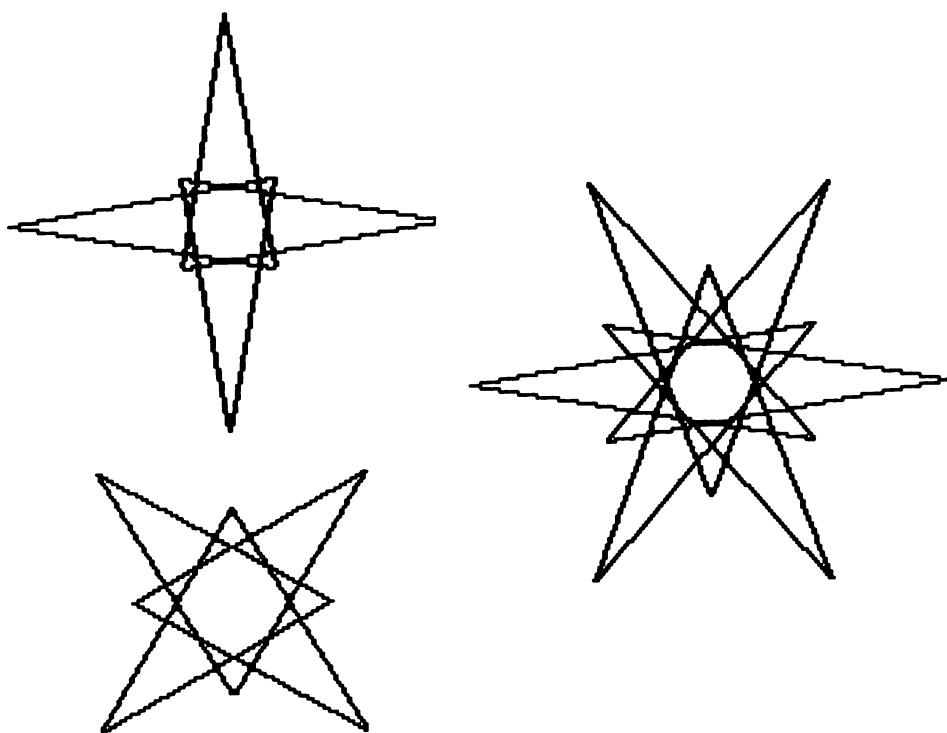



Figura 8.21 Disegni tracciati con POLANZAGAN

Tutte queste variazioni possono essere applicate anche alle spirali segmentate e alle volute, semplicemente aumentando (o riducendo) l'angolo di rotazione ad ogni passo.
Altre figure divertenti si costruiscono partendo da una spirale segmentata, che però si arresta dopo un certo numero di passi e viene ripetuta per un dato numero di volte; in Figura 8.22 si vede una spirale con angolo di 90 gradi tracciata per sette tratti, e ripetuta continuamente:

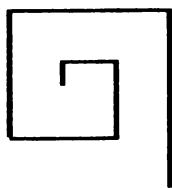
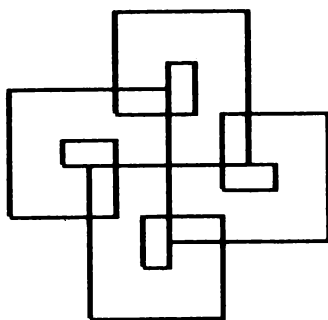
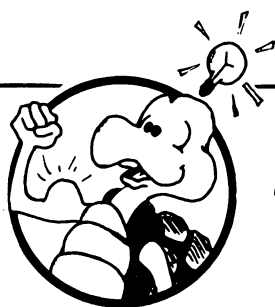
```
PER SPIROVOLTE :GRANDEZZA :ANGOLO :NUMERO
SE :NUMERO = 0 STOP
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SPIROVOLTE (:GRANDEZZA+10) :ANGOLO (:NUMERO-1)
FINE
PER SPIMOSAICO :ANGOLO :NUMERO
SPIROVOLTE 10 :ANGOLO :NUMERO
SPIMOSAICO :ANGOLO :NUMERO
FINE
```

SPIROVOLTE 10 90 7 SPIMOSAICO 90 7

Figura 8.22 Variazioni su un tema di spirale

IDEA

SPIROVOLTE è una procedura che conta. È molto simile alla SPIRASEG (del paragrafo 8.4), solo che ha in più la variabile :NUMERO, che indica esattamente quanti tratti vanno disegnati. Ad ogni SPIROVOLTE che viene chiamata, il valore :NUMERO viene diminuito di uno, e quando diventa zero la procedura si arresta. La SPIMOSAICO invece continua a richiamare la SPIROVOLTE assegnando il valore fisso 10 a :GRANDEZZA.

Queste spiraloidi possono essere anche tracciate con grandezza a piacere, se si rende variabile in SPIROVOLTE l'incremento della lunghezza ad ogni tratto, e si assegna ad una variabile di SPIMOSAICO la lunghezza del tratto di partenza.
Ecco:

```
PER SPIVOLTEVAR :GRANDEZZA :ANGOLO :AUMEN :NUMERO
SE :NUMERO = 0 STOP
AVANTI :GRANDEZZA
DESTRA :ANGOLO
SPIVOLTEVAR (:GRANDEZZA+:AUMEN) :ANGOLO :AUMEN
   (:NUMERO-1)
FINE
PER SPISAICOVAR :GRANDEZZA :ANGOLO :NUMERO
SPIVOLTEVAR :GRANDEZZA :ANGOLO :GRANDEZZA :NUMERO
SPISAICOVAR :GRANDEZZA :ANGOLO :NUMERO
FINE
```

Quando SPISAICOVAR richiama SPIVOLTEVAR, assegna lo stesso valore a :GRANDEZZA e a :AUMEN di quest'ultima, così il disegno mantiene le figure in proporzione fra loro.
Le spiraloidi sono un argomento matematico affascinante, anche perché si comportano in modo molto vario: ad esempio, alcuni tipi hanno forma

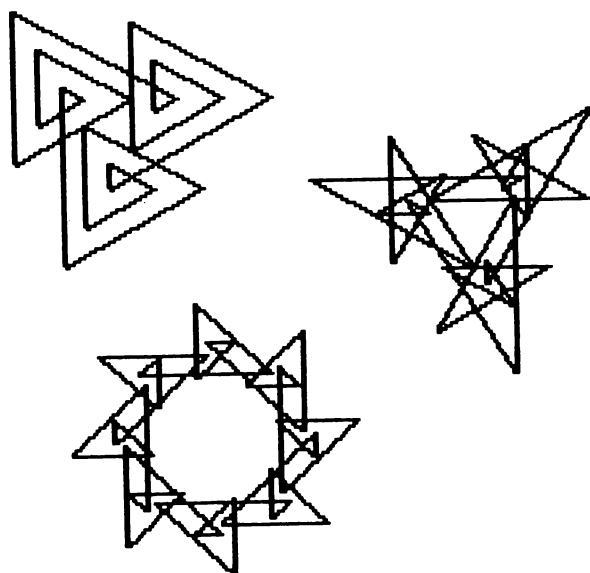



Figura 8.23 Altre spiraloidi

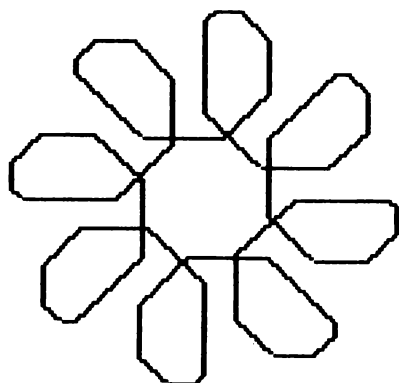
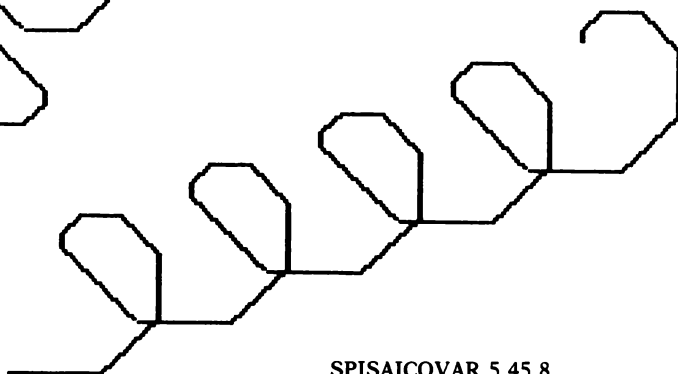



Figura 8.24 Spiraloide aperta e spiraloide chiusa

chiusa, mentre altri continuano ad avvolgersi attorno allo schermo senza fine.

Quali combinazioni di angoli e grandezze producono forme chiuse? E quali tracciano forme aperte, che non ripassano mai più dal punto di partenza? La risposta è molto simile alla regola che riguarda il numero di lati tracciati da POLIGONALE e l'angolo assegnato, vista nel paragrafo 8.1.

L'enfasi di questo capitolo è posta sulla costruzione di disegni e sull'esplorazione della loro struttura. Forse qualcuno preferirebbe un approccio più sistematico, volto ad insegnare alcune proprietà matematiche molto interessanti delle figure qui presentate. Certo, gli aspetti matematici delle graziose

figure viste in questo capitolo dovrebbero far parte del curriculum scolastico di ogni studente, e credo che facendo ciò si perderebbe forse parte della loro poesia.
Chi volesse usare questo libro come occasione di apprendimento della matematica, dovrebbe tener conto di alcuni suggerimenti che compaiono qui di seguito.
Per prima cosa, conviene insistere sulla scoperta di motivi costruiti combinando insieme forme elementari: in questi esercizi, bisogna fare attenzione alla definizione degli angoli, che sono il fattore critico per la determinazione della forma finale.
C'è poi da tener sempre presente la difficoltà che molti hanno, di considerare più variabili nello stesso tempo. La cosa migliore è certamente procedere con metodo, isolando l'effetto di una variabile alla volta: l'esperienza insegna però che questo approccio sistematico viene rifiutato da molti allievi che, almeno all'inizio, preferiscono partire dal tutto, invece che dalle parti. Molti insegnanti non tengono conto di questa tendenza degli allievi, oppure cercano di contrastarla come se fosse una deviazione dannosa, imponendo un lavoro di "ricerca razionale" prima che gli allievi siano pronti, con la mente e con lo spirito, a seguirli su questa strada, che può anche sembrare arida e noiosa, a prima vista. Bisogna fare attenzione: forzando prematuramente gli allievi alla ricerca metodica si spreca tempo e si rischia addirittura di far nascere in molti un sentimento di avversione verso la matematica; se si sa invece aspettare il momento giusto, gli allievi non mancheranno di apprezzare il nuovo metodo, e anzi diventeranno presto entusiasti.
Insomma, l'insegnante dovrebbe incoraggiare lo sviluppo del pensiero critico, senza inibire il comportamento esplorativo e il pensiero creativo. Occorre sensibilità, per accorgersi di quali consigli ha bisogno, in un certo momento, l'allievo, e di cosa veramente sta cercando di ottenere: del resto, non è forse l'insegnamento più un'arte che una scienza?
In effetti, questo capitolo è stato scritto con lo stesso spirito, rimanendo in bilico fra matematica e disegno, fra indagine critica e esplorazione creativa, lasciando a chi legge la libertà di trovare ciò che al momento più gli serve. La predica è finita!

Dialogare col computer

Questo capitolo si interessa del modo in cui il Logo memorizza le informazioni, e spiega come si fa a dialogare con il computer. Vengono presentati i tre tipi di informazioni trattati dal Logo: numeri, parole e liste. Leggendo il capitolo, diventa più facile capire come i computer elaborano i dati (si dice così) in loro possesso. La tecnica dell'elaborazione automatica dei dati riguarda infatti il modo in cui il computer riceve, memorizza, modifica e stampa le informazioni: spesso tutto ciò richiede l'interazione del computer e della persona che lo usa.
Qui si presentano, appunto, procedure interattive scritte in Logo.

Comando	Abbreviazione	Esempi
		STAMPA 360 / 3, DESTRA 360 / 3
PAROLA		STAMPA PAROLA "ASTRO "NAVE
FRASE	FR	STAMPA FRASE [BENVENUTO] [AMICO]
		STAMPA FRASE "EHI [AMICO]
		STAMPA FRASE "EHI "AMICO
		STAMPA (FR [EHI] [CARO] [AMICO]
PRIMO	PRI	STAMPA PRIMO "EHI
		STAMPA PRIMO [EHI CARO AMICO]
MENPRI	MP	STAMPA MENPRI "EHI
		STAMPA MP [EHI CARO AMICO]
ULTIMO	ULT	STAMPA ULTIMO "EHI
		STAMPA ULTIMO [EHI CARO AMICO]
MENULT	MU	STAMPA MU "EHI
LEGGIRIG	LR	ASSEGNA "RISPOSTA LEGGIRIG
PULISCITESTO		

Comando	Abbreviazione	Esempi
SCRIVI		SCRIVI [INDOVINA UN NUMERO]
ACASO		STAMPA ACASO 20
VERIFICA		VERIFICA :RISPOSTA = 7
SEVERO	SEV	SEVERO STAMPA [INDOVINATO!]
SEFALSO	SEF	SEFALSO STAMPA [HAI SBAGLIATO...]

Vengono usati i file che si trovano nell'Appendice A "LEGGINUMERO, "INDOVINUMERO e "MATQUIZ.
Vengono usate nuove procedure di servizio:

Nome	Esempio
LEGGINUMERO	ASSEGNA "RISPOSTA LEGGINUMERO

9.1 Numeri, parole e liste

C'è un nome generale per qualunque informazione data al computer: dato. Si dice "un dato" quando non si vuole (o non si sa) distinguere il tipo di informazione di cui si parla.
Il Logo può trattare tre tipi di dato: numeri, parole e liste.
Per il computer, i numeri sono quelli che sono per noi: grandi e piccoli, positivi e negativi. Con essi, vengono eseguite le normali operazioni aritmetiche (addizione, sottrazione, divisione e moltiplicazione). Il computer è anche capace di confrontare due numeri e di dire se sono uguali o no, e qual è allora il più grande dei due (o il più piccolo).
Ecco alcuni esempi da provare:

```
STAMPA 4+5
STAMPA 36-6
STAMPA 36/7
STAMPA (3+5)*7
```

Le parentesi che racchiudono numeri, come in (3+5) sono un segno importante per il computer, che esegue prima le operazioni interne (in questo caso, la somma 3+5 ottenendo 8) e poi quelle esterne, con il risultato appena ottenuto (nell'esempio, 8*7). Se non si battono le parentesi, il computer si comporta diversamente:

```
STAMPA 3+5*7
```

In questo caso il Logo prima moltiplica 5 e 7, e poi somma 3 al risultato. Se non ci sono parentesi, il Logo dà sempre la precedenza alle moltiplica-

zioni e alle divisioni, ed esegue per ultime le addizioni e le sottrazioni. Ecco altri esempi:

```
STAMPA 5=7
STAMPA 5=(7-2)
STAMPA 5>7
STAMPA 5<7
```

Per il computer, l'espressione 5>7 è una domanda, e significa: "È vero che 5 è maggiore di 7?". Il Logo, naturalmente, in questo caso risponde FALSO. L'espressione 5<7, invece, domanda al Logo: "È vero che 5 è minore di 7?" (il Logo risponde VERO, in questo caso).
Nel secondo esempio, le parentesi (7−2) non sono necessarie, perché il Logo esegue sempre tutte le operazioni matematiche prima di passare alle operazioni di confronto: certe volte però è meglio battere lo stesso le parentesi, perché così si riesce a leggere meglio le istruzioni che si sono scritte.
Le parole, per il Logo, sono quasi come per noi le parole della nostra lingua. In effetti, il Logo accetta come parola quasi ogni possibile combinazione di lettere, numeri e simboli (o caratteri, come anche si dice). Per distinguere una parola, la si fa incominciare con il simbolo " (le virgolette): attenzione però a non far comparire il simbolo " all'interno di una parola, o alla fine, perché allora il Logo si confonde.
Ecco una differenza fra il Logo e la nostra lingua (e molti altri linguaggi di computer): quasi ogni possibile combinazione di simboli può formare una parola. Ecco qualche caso:

```
STAMPA "BUONGIORNO
STAMPA "ABCXYZ
STAMPA "R2D2
STAMPA "AB.$-)**
STAMPA "3+4
```

A cosa servono le parole del Logo?
Intanto, per i nomi delle variabili e dei file: in effetti, abbiamo usato Logoparole fin dal Capitolo 4.
Anche i numeri possono essere parole, per il Logo; come si vede, possono essere usati con e senza il simbolo ":

```
STAMPA "25
STAMPA "25+"25
```

Se si lascia il simbolo " da solo, si ha una cosiddetta parola vuota, che può servire ad esempio per produrre una riga vuota:

STAMPA "

Una Logoparola termina sempre con uno spazio bianco, e solo con questo. Proviamo ora questo caso:

STAMPA "BUONGIORNO AMICO

Il Logo stampa la parola "BUONGIORNO, ma poi prende AMICO per una procedura, perché non incomincia con le virgolette. Se ora si batte:

STAMPA "BUONGIORNO "AMICO

il Logo riconosce la parola "AMICO, che si presenta per seconda, ma non sa che cosa farne.
Le cosiddette liste sono il modo in cui il Logo combina le parole in gruppi. La lista, si distingue perché è racchiusa fra parentesi quadre, e può comprendere parole, numeri e persino altre liste.
Ecco i primi esempi:

```
STAMPA [BUONGIORNO AMICO]
STAMPA [1 2 3 4 5 6]
STAMPA [MI CHIAMO RICCARDO]
STAMPA [ECCO UNA LISTA: [ECCO UNA LISTA]]
STAMPA [[A B] [C D] [E F]]
```

Una lista che non contiene simboli, come [ ], è detta lista vuota, e può ad esempio servire anch'essa, come la parola vuota, per stampare una linea vuota:

STAMPA [ ]

9.2 Comandi per l'uso di parole e liste

I comandi del Logo consentono di combinare le parole fra di loro, formando parole più lunghe oppure liste, e di combinare parole e liste insieme, formando nuove liste. È anche possibile produrre l'effetto opposto, separando in parti parole e liste.
Il comando PAROLA serve per combinare in una sola parola due parole separate, che sono i suoi dati di input:

```
STAMPA PAROLA "ASTRO "NAVE
STAMPA PAROLA "PAR "OLA
```

Il comando PAROLA accetta anche più di due parole come dati di input, ma allora il comando e i dati vanno scritti fra parentesi tonde, come in questo esempio:

STAMPA (PAROLA "PALLA "CANES "TRO)

TRABOCCHETTO

Attenti a lasciare lo spazio vuoto dopo l'ultima parola, prima della parentesi), perché altrimenti il Logo prende la parentesi come un simbolo dell'ultima parola.
Provate a battere:

STAMPA (PAROLA "PALLA "CANES "TRO)

Un altro bug molto comune è aprire la parentesi dopo il comando PAROLA:

STAMPA PAROLA ("PALLA "CANES "TRO)

Anche in questo caso il Logo protesta.

Con il comando FRASE si riesce a combinare parole e liste, per formare una nuova lista, più grande. Di solito FRASE ha due dati di input, ciascuno dei quali può essere una parola o una lista.
Ecco qualche esempio:

STAMPA FRASE "UNA [PAROLA E UNA LISTA]
STAMPA FRASE [UNA PAROLA E UNA] "LISTA
STAMPA FRASE "DUE "PAROLE

Vale anche l'abbreviazione FR:

STAMPA FR [DA DUE LISTE] [SE NE FA UNA]

Il comando FRASE accetta più di due dati di input, se tutto quanto (comando compreso) viene racchiuso fra parentesi.
Ecco l'esempio:

```
STAMPA (FRASE "QUESTA [DIVENTA UNA] "SOLA "LISTA)
```

I comandi PRIMO (o PRI), ULTIMO (o ULT), MENPRI (o MP) e MENULT (o MU) servono per separare parole o liste. Ecco alcuni esempi (se ne possono trovare altri da soli):

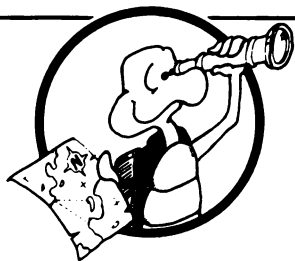
```
STAMPA PRIMO "BUONGIORNO
STAMPA ULTIMO "BUONGIORNO
STAMPA MENPRI "BUONGIORNO
STAMPA MENULT "BUONGIORNO
```

PRIMO fornisce il primo carattere di una parola (e ULTIMO, l'ultimo carattere). MENPRI invece funziona al contrario, fornendo tutti i caratteri della parola data in input, meno il primo (MENULT fornisce invece la parola di input, eccetto il suo ultimo carattere). Questi comandi si applicano anche a una lista, oltre che a una parola. Ecco alcuni esempi:

```
STAMPA PRIMO [BUONGIORNO CARO AMICO]
STAMPA ULTIMO [BUONGIORNO CARO AMICO]
STAMPA MENPRI [BUONGIORNO CARO AMICO]
STAMPA MENULT [BUONGIORNO CARO AMICO]
```

PRIMO fornisce il primo elemento della lista data in input (di solito, è una parola). ULTIMO estrae invece l'ultimo elemento della lista data. MENPRI e MENULT producono invece la lista di input, meno il primo elemento (o l'ultimo, rispettivamente). Questi comandi si possono combinare in vario modo, e producono effetti notevoli. Ecco qualche esempio:

```
STAMPA PRIMO MENPRI [BUONGIORNO CARO AMICO]
STAMPA MENPRI PRIMO [BUONGIORNO CARO AMICO]
```

Si riesce a capire cosa fanno? Ecco qua: MENPRI della lista [BUONGIORNO CARO AMICO] è la lista [CARO AMICO]; PRIMO di questa lista è la parola "CARO. Perciò il primo esempio produce la parola "CARO. PRIMO della lista [BUONGIORNO CARO AMICO] è la parola "BUONGIORNO; MENPRI di "BUONGIORNO è la parola "UONGIORNO, che è quindi il risultato del secondo esempio.

ESPLORAZIONE

È bene provare per bene i comandi PRIMO, ULTIMO, MENPRI e MENULT. Come si è visto, la combinazione PRIMO ULTIMO [BUONGIORNO CARO AMICO] produce la seconda parola della lista: si riesce a trovare una combinazione di comandi che produce la terza parola della lista di input? E la quarta? E, perché no, la penultima, o la terzultima?

Il comando LEGGIRIG fa attendere il Logo, finché non viene battuta una riga da tastiera, e poi fornisce la stessa riga in forma di lista. Ecco qui un piccolo divertimento che usa i due comandi FRASE e LEGGIRIG:

```
PER PARLA
STAMPA [PER FAVORE, SCRIVIMI QUALCOSA DA DIRE]
STAMPA FRASE [MI FAI DIRE] LEGGIRIG
FINE
```

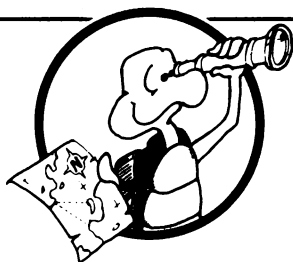
Quando il Logo incontra il comando LEGGIRIG, si ferma e aspetta che si batta qualcosa alla tastiera, poi scrive una frase che comprende ciò che si è appena battuto. Provate PARLA per un po'.
Ecco una variazione:

```
PER RIBATTI
STAMPA [PER FAVORE, SCRIVIMI QUALCOSA DA DIRE]
STAMPA FRASE [MA IO NON VOGLIO DIRE] LEGGIRIG
RIBATTI
FINE
```

Provate anche RIBATTI (per fermarla, bisogna battere CTRL-G).
Eccone un'altra:

```
PER GUSTI
STAMPA [PER FAVORE, SCRIVIMI QUALCOSA CHE TI PIACE]
STAMPA FRASE LEGGIRIG [PIACE ANCHE A ME]
STAMPA [E COSA NON TI PIACE?]
STAMPA (FRASE [A ME] LEGGIRIG [PIACE ANCORA MENO!])
```

GUSTI
FINE

Si noti che, in un caso, si usano le parentesi con il comando FRASE, dato che ci sono tre dati di input.

ESPLORAZIONE

Conviene scrivere da soli qualche procedura di tipo PARLA. Si riesce a prepararne una che non è d'accordo con nulla?

SUGGERIMENTO

La seconda istruzione di PARLA merita un interesse speciale. Essa comprende ben tre comandi del Logo e due liste di dati:

```
STAMPA FRASE [MI FAI DIRE] LEGGIRIG
```

Il comando STAMPA richiede dati da stampare, e se li procura dal comando FRASE. A sua volta, FRASE ha bisogno di due dati di input, che possono essere due parole, due liste, o anche una parola e una lista. Nella PARLA, è il comando STAMPA stesso che fornisce a FRASE un dato di input, la lista [MI FAI DIRE], e che indica di richiamare un terzo comando, LEGGIRIG, per il secondo dato.
La LEGGIRIG prende come input tutto quello che viene battuto alla tastiera, e lo fornisce in output (si dice così) in forma di lista, a disposizione della procedura che l'ha richiamata: in questo caso, supponiamo di battere BRAVO.

La FRASE, allora, combina BRAVO con il suo primo dato di input, e fornisce in output alla STAMPA la lista [MI FAI DIRE BRAVO].
La STAMPA finalmente fa comparire sullo schermo TV la lista MI FAI DIRE BRAVO (senza le parentesi quadre).
Qui si è introdotto un nuovo termine del gergo informatico: output. Si parla di output per indicare il prodotto di una procedura (in genere, un dato che viene poi passato ad un'altra procedura, che lo usa come input, cioè come dato assegnato). Si può pensare all'output anche come ad un messaggio, che una procedura invia in risposta al richiamo da parte di un'altra, o di un comando. Nel nostro esempio, l'output di FRASE è l'input di STAMPA, che lo passa allo schermo TV.
I comandi e le procedure che producono dati in output prendono anche il nome di operazioni, per analogia con le normali operazioni aritmetiche (che forniscono un output numerico) e con quelle di confronto (proprie dei comandi <, > e =) che forniscono in output la parola "VERO o la parola "FALSO. Anche PRIMO, ULTIMO, MENPRI e MENULT possono essere considerati operazioni, il cui output è fatto da parole o da liste.

9.3 Numeri, parole e liste come variabili

Abbiamo già visto, nei Capitoli 7 e 8, come i numeri possono essere assegnati a variabili che controllano la grandezza o la forma di un disegno. Anche le parole o le liste possono rappresentare il valore di una variabile, e possono essere assegnate ad esempio dandole in input ad una procedura:

```
PER DIMMI :MESSAGGIO
STAMPA [TI DICO]
STAMPA :MESSAGGIO
FINE
```

L'input di DIMMI può essere una parola, una lista, o anche un numero. Si può battere, ad esempio:

```
DIMMI "BUONGIORNO
DIMMI [È ORA DI ANDARE]
DIMMI 33
```

Un altro modo di indicare un numero, una parola o una lista come valore di una variabile ricorre al comando ASSEGNA, che ha due input: il nome della variabile (che deve essere una Logoparola) e il valore che si intende assegnare (che può essere, appunto, un numero, una parola o una lista). Supponiamo di battere:

ASSEGNA "MESSAGGIO [CHE PASTICCIO]

Il comando ASSEGNA prepara una nuova variabile, chiamata "ASSEGNA, e vi associa il valore [CHE PASTICCIO].
Il Logo prende la nuova variabile (che è come una casella, ricordate...) con il suo valore e la sistema nella memoria di servizio del computer (che è come un grande archivio). Per vedere il nuovo valore, basta battere:

STAMPA :MESSAGGIO

Quando si modifica il valore della variabile, ASSEGNA crea una nuova variabile, con lo stesso nome (nel nostro esempio, "MESSAGGIO) e con il nuovo valore, ad esempio "BUONGIORNO:

ASSEGNA "MESSAGGIO "BUONGIORNO

Il Logo sistema questo nuovo valore nella memoria del computer, eliminando il vecchio. Per vedere il nuovo valore, basta battere ancora:

STAMPA :MESSAGGIO

Non c'è limite alle modifiche. Si può battere:

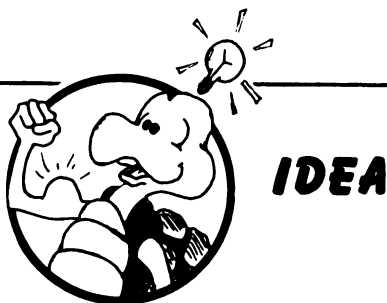
ASSEGNA "MESSAGGIO 1000

e vedere il nuovo valore battendo:

STAMPA :MESSAGGIO

... e così via ...
Ecco una prova da fare (indovinate cosa produce?):

STAMPA "MESSAGGIO
STAMPA :MESSAGGIO

STAMPA "MESSAGGIO dice al computer di stampare la parola "MESSAGGIO. Il comando STAMPA :MESSAGGIO fa invece stampare il valore della variabile chiamata "MESSAGGIO. È importante distinguere questi due concetti, e non bisogna mai dimenticare le virgolette o i due punti, quando si scrive il comando o quando magari lo si detta ad altri. Anzi, è meglio dire addirittura a tutte lettere questi due simboli, ad esempio ... virgolette MESSAGGIO, oppure ... due punti MESSAGGIO.

La trovata di assegnare un nome a un dato, e poi di usare questo nome nelle istruzioni che indicano come elaborare il dato stesso, è una delle più importanti nel campo della matematica e dell'informatica. È molto importante però capire bene la differenza fra il dato e il suo nome: ecco perché sono importanti le virgolette e i due punti! Con il loro impiego, non ci sono malintesi.

Il comando ASSEGNA può essere usato anche per modificare il valore di una variabile:

```
ASSEGNA "NUMERO 5
STAMPA :NUMERO
ASSEGNA "NUMERO :NUMERO+5
STAMPA :NUMERO
```

Nell'esempio qui sopra, il comando ASSEGNA "NUMERO :NUMERO+5 indica al computer di sommare 5 al valore :NUMERO, e di assegnare il risultato (o come anche si dice in gergo, l'output dell'operazione) alla stessa variabile chiamata "NUMERO.
Con ASSEGNA si può cambiare il valore anche a variabili di tipo parola, o di tipo lista:

```
ASSEGNA "MESSAGGIO [BUONGIORNO CARO]
STAMPA :MESSAGGIO
ASSEGNA "MESSAGGIO FRASE :MESSAGGIO "AMICO
STAMPA :MESSAGGIO
```

SUGGERIMENTO

Consideriamo il comportamento del comando:

ASSEGNA "MESSAGGIO FRASE :MESSAGGIO "AMICO

Per prima cosa, ricordiamo che con il primo comando dell'esempio appena proposto, ASSEGNA "MESSAGGIO [BUONGIORNO CARO], si crea una variabile chiamata "MESSAGGIO il cui valore è la lista [BUONGIORNO CARO], che viene memorizzata dal Logo nella memoria di servizio del computer.
Con il comando STAMPA :MESSAGGIO, si fa cercare al Logo la casella chiamata "MESSAGGIO, il cui contenuto [BUONGIORNO CARO] viene passato a STAMPA.
Arriviamo finalmente all'istruzione che ci interessa:

ASSEGNA "MESSAGGIO FRASE :MESSAGGIO "AMICO

Anche qui, come al solito, ASSEGNA deve avere due dati in input, un nome e un valore da associare a questo. Il primo dato, il nome, è "MESSAGGIO. Per il secondo, il Logo indica ad ASSEGNA di richiamare FRASE, fornendogli naturalmente i dati di cui questo comando ha bisogno, a sua volta, che in questo caso sono :MESSAGGIO e "AMICO.
A sua volta, ASSEGNA entra in azione e richiama FRASE, fornendo i due dati di input che occorrono: il primo è il valore che al momento è memorizzato nella casella chiamata "MESSAGGIO, cioè ancora la lista [BUONGIORNO CARO]; il secondo dato è la parola "AMICO.
FRASE fa la sua parte, combinando i due dati di input e restituendo ad ASSEGNA la lista [BUONGIORNO CARO AMICO], che serve a quest'ultimo come secondo dato di input.
Il comando ASSEGNA crea ora una nuova variabile, chiamata "MESSAGGIO, che vale la lista [BUONGIORNO CARO AMICO]. Il Logo la prende e la memorizza al posto della vecchia variabile, che viene eliminata.
Se si batte, a questo punto, il comando STAMPA :MESSAGGIO, il Logo fornisce al comando STAMPA il nuovo valore.

Ecco un altro esempio di come si può impiegare il comando FRASE in una procedura:

```
PER CRESCE :MESSAGGIO
STAMPA [DIMMI QUALCOSA DI NUOVO]
```

```
ASSEGNA "NUOVAPARTE LEGGIRIG
ASSEGNA "MESSAGGIO FRASE :MESSAGGIO :NUOVAPARTE
STAMPA [E IO TI DICO]
STAMPA :MESSAGGIO
CRESCE :MESSAGGIO
FINE
```

Provate a vedere cosa succede battendo CRESCE :BUONGIORNO seguendo le indicazioni del computer.
Ecco come si comporta CRESCE: la prima istruzione fa scrivere DIMMI QUALCOSA DI NUOVO. La seconda assegna il nome "NUOVAPARTE a ciò che viene battuto alla tastiera. La terza istruzione costruisce una lista con :MESSAGGIO e :NUOVAPARTE, chiamandola ancora "MESSAGGIO. La quarta istruzione visualizza la scritta E IO TI DICO. La quinta stampa la lista appena costruita, e la sesta richiama una nuova CRESCE, con il nuovo :MESSAGGIO come dato di input, che si comporta nello stesso modo, e così via.

9.4 Domande e risposte

Ora possiamo preparare alcuni semplici giochi di domanda e risposta. Eccone uno:

QUIZ1	si richiama il gioco
CHE TITOLO HA QUESTO LIBRO?	domanda del computer
LOGO	prima risposta
NO...RIPROVA...	la risposta non va bene
CHE TITOLO HA QUESTO LIBRO?	il gioco prosegue ...
IMPARO IL LOGO	altra risposta
NO...RIPROVA...	ancora sbagliata ...
CHE TITOLO HA QUESTO LIBRO?	il gioco prosegue ...
IL LOGO PER IL COMMODORE 64	altra risposta ...
RISPOSTA ESATTA!	il gioco termina

Come si fa a preparare il QUIZ1? Non è difficile: basta applicare le idee appena viste:

```
PER QUIZ1
STAMPA [CHE TITOLO HA QUESTO LIBRO?]
ASSEGNA "RISPOSTA LEGGIRIG
SE :RISPOSTA = (IL LOGO PER IL COMMODORE 64) STAMPA
    [RISPOSTA ESATTA!] STOP
```

```
STAMPA [NO...RIPROVA...]
QUIZ1
FINE
```

Si noti che a volte, come in questo caso, bisogna andare a capo con un'istruzione particolarmente lunga (si batte il tasto RETURN solo alla fine).
La QUIZ1 non si ferma da sola (bisogna arrestarla con CTRL-G).
Ecco un altro esempio:

```
PER QUIZ2
STAMPA [DIMMI L'ABBREVIAZIONE DI AVANTI...]
ASSEGNA "RISPOSTA LEGGIRIG
SE :RISPOSTA = [A] STAMPA [ESATTO!] STOP
STAMPA [LA RISPOSTA E' SBAGLIATA]
STAMPA [VUOI RIPROVARE?]
ASSEGNA "RIPROVO LEGGIRIG
SE :RIPROVO = [NO] STAMPA [DOVEVI DIRE: A] STOP
SE :RIPROVO = [SI] QUIZ2 STOP
STAMPA [NON SCEGLI NE' NO NE' SI. ABBANDONO]
FINE
```

Come si vede, la QUIZ2 è più completa di QUIZ1, perché fa scegliere se proseguire o no il gioco (se non si risponde SI o NO, la procedura si arresta senza dare la risposta giusta).

SUGGERIMENTO

In QUIZ2 c'è una novità che è l'impiego del comando STOP alla fine di certe istruzioni. La situazione delle righe 3 e 7 è chiara: se viene data la risposta giusta, o se chi gioca vuole smettere, la procedura si deve arrestare. Però cosa ci fa il comando STOP alla fine dell'istruzione 8? Lì viene richiamata una nuova procedura QUIZ2 e il gioco prosegue: perché allora indicare STOP?
Bisogna sapere che il comando STOP arresta solo la procedura a cui appartiene, mentre se questa è stata richiamata da un'altra, quella prosegue.
Per capire a cosa serve quello STOP, basta provare ad eliminarlo (con il co-

mando EDITA) e a giocare con QUIZ2 così modificata. Se si risponde apposta per qualche volta in modo sbagliato, e poi si risponde NO alla domanda se si vuole riprovare, succede questo:

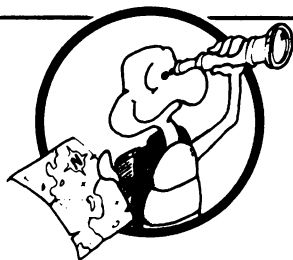
```
QUIZ 2
DIMMI L'ABBREVIAZIONE DI AVANTI
AVA
LA RISPOSTA E' SBAGLIATA
VUOI RIPROVARE?
SI
DIMMI L'ABBREVIAZIONE DI AVANTI
AV
LA RISPOSTA E' SBAGLIATA
VUOI RIPROVARE?
NO
DOVEVI DIRE: A
NON SCEGLI NE' NO NE' SI. ABBANDONO
```

Ma l'ultima riga è fuori posto! Se si risponde NO il gioco deve terminare con la risposta giusta e basta!
Il fatto è che nell'ultimo esempio ci sono due QUIZ2 attive, contemporaneamente (la prima aspetta che la seconda sia terminata). La seconda QUIZ2 parte nel momento in cui si risponde SI alla domanda se si vuole riprovare: quando poi si risponde NO, che non si vuole riprovare per la terza volta, la seconda QUIZ2 si arresta, ma rimane ancora la prima, che prosegue per la sua strada, scrivendo il messaggio che qui è fuori posto. Se ci fosse STOP dopo il richiamo di QUIZ2 nell'istruzione 8 (come nella versione originale), la QUIZ2 riprenderebbe con questo e si arresterebbe subito: se invece questo STOP non c'è, la QUIZ2 che rimane salta all'istruzione successiva e stampa quello che si è visto (poi si arresta perché non trova più altri comandi).
Questo tipo di bug può essere evitato se si colloca l'istruzione recursiva per ultima, come in QUIZ1: quando si ha una procedura un po' complessa, bisogna però sempre fare attenzione di aver collocato tutti gli STOP che occorrono.
Per scovare i bug di questo tipo è utile a volte recitare dal vivo, con un gruppo di persone, la procedura che interessa: si dà a tutti il copione (cioè l'elenco delle istruzioni) e si fa attenzione che ciascuno prosegua con la sua parte, quando è finita quella che lo faceva stare in attesa.

La QUIZ2 si può ancora migliorare, specie per tener conto che la risposta giusta, spesso, può essere data in modi diversi: in effetti, le procedure di questo tipo devono tener conto di ogni possibile modo di dare la risposta corretta. Torna utile allora il comando UNOVERO? che fornisce in output il valore "VERO quando uno almeno dei suoi dati di input vale "VERO (il Logo ha anche il comando TUTTIVERI? il cui output vale "VERO solo quando tutti i suoi dati di input valgono "VERO).

Ecco:

```
PER QUIZ3
STAMPA [QUANTI DATI DI INPUT]
STAMPA [VUOLE IL COMANDO ASSEGNA?]
ASSEGNA "RISPOSTA LEGGIRIG
SE UNOVERO? :RISPOSTA = [DUE] :RISPOSTA = [2] STAMPA [OK!]
   STOP
STAMPA [NO...VUOLE DUE DATI DI INPUT,]
STAMPA [UN NOME E UN VALORE]
FINE
```

Come si vede, QUIZ3 tiene conto dei due possibili modi di dare la risposta giusta, anche se non prevede che il gioco prosegua con altri tentativi.

ESPLORAZIONE

Perché non provare a costruire altri quiz da soli? Si può anche, volendo, preparare un superprocedura di quiz, combinando quelle di cui si dispone:

```
PER LOGOQUIZ
QUIZ1
QUIZ2
QUIZ3
QUIZ4
QUIZ5
.........
FINE
```

Il quiz è una buona occasione per imparare a scrivere procedure di un certo impegno. C'è un sacco di cose da fare! Come visualizzare le domande sullo schermo? Che tono dare alle domande e alle risposte (serio, scherzoso, amichevole ...) e quante risposte sbagliate si tollerano prima di terminare il gioco e dare la risposta giusta? Conviene far dare dei suggerimenti dal computer? E quando il computer comunica che una risposta è sbagliata, può dire in che cosa sbaglia? Vale la pena che il computer

tenga il conto del numero di risposte errate? E di quelle giuste? Eh sì, una procedura di quiz "intelligente" è proprio un bell'impegno, perché deve tener conto di molte possibilità di scelta: alla fine, ci si può trovare con un programma molto, molto complesso.

9.5 Giochiamo a INDOVINUMERO?

Con INDOVINUMERO, il gioco sta nel cercare di indovinare il numero scelto a caso dal computer:

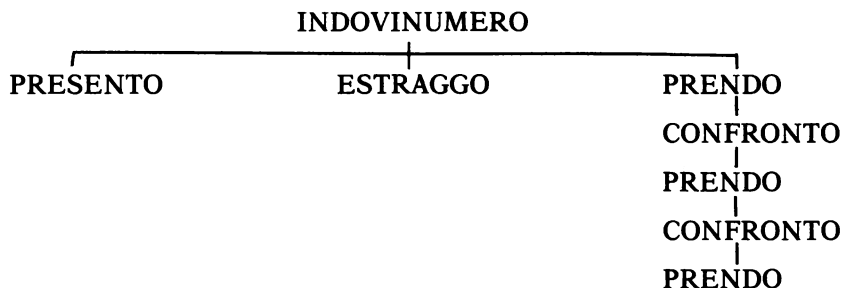
```
INDOVINUMERO
STO PENSANDO A UN NUMERO FRA 0 E 100!
INDOVINA QUAL E'?
>50
NO: AUMENTA
>75
NO: AUMENTA
>85
NO: CALA
>80
NO: AUMENTA
>82
NO: CALA
>81
INDOVINATO!
```

INDOVINUMERO è già pronto, basta copiare le istruzioni riportate qui sotto (attenzione, perché viene usata la procedura LEGGINUMERO, che fa sempre parte dell'Appendice A).
Con INDOVINUMERO si ha più o meno la stessa situazione vista con QUIZ, solo che ora il computer indica il motivo dell'errore, e guida così chi gioca verso la risposta esatta. Vediamo che cosa deve fare il computer per giocare a INDOVINUMERO:

1. Visualizzare un messaggio di presentazione del gioco.
2. Scegliere a caso un numero fra 0 e 100 e tenerlo segreto.
3. Attendere che il giocatore batta un numero.
4. Verificare se il numero battuto alla tastiera è giusto, o è troppo grande, oppure troppo piccolo. Negli ultimi due casi, deve visualizzare un messaggio e attendere il nuovo tentativo.

Bene, ognuna delle funzioni elencate sopra è affidata a una procedura distinta:

```
PER INDOVINUMERO
PRESENTO
ESTRAGGO
PRENDO
FINE
PER PRESENTO
PULISCITESTO
STAMPA [STO PENSANDO A UN NUMERO FRA 0 E 100!]
STAMPA [INDOVINA QUAL E'?]
FINE
PER ESTRAGGO
ASSEGNA "NUMERO 1+ACASO 99
FINE
PER PRENDO
SCRIVI ">
ASSEGNA "TENTATIVO LEGGINUMERO
CONFRONTO :TENTATIVO :NUMERO
FINE
PER CONFRONTO :TENTATIVO :NUMERO
SE :TENTATIVO = :NUMERO STAMPA [INDOVINATO!] STOP
SE :TENTATIVO > :NUMERO STAMPA [NO: CALA] PRENDO STOP
SE :TENTATIVO < :NUMERO STAMPA [NO: AUMENTA]
   PRENDO STOP
FINE
```

Vale la pena di considerare lo schema di connessione fra le varie procedure (è questo un modo molto utile per capire bene il comportamento di programmi complessi).
La INDOVINUMERO richiama PRESENTO, ESTRAGGO e PRENDO; questa, da parte sua, richiama LEGGINUMERO e CONFRONTO. CONFRONTO, poi, richiama un'altra PRENDO. Allora CONFRONTO e PRENDO continuano a richiamare l'una nuovi replicanti dell'altra, finché il numero non viene indovinato.
Ecco lo schema:

Come si vede, lo schema assomiglia ad un albero stilizzato e capovolto. I rami più bassi continuano ad estendersi fino a che il numero non viene indovinato.
In INDOVINUMERO sono impiegati tre comandi del Logo mai visti finora, PULISCITESTO, ACASO e SCRIVI, oltre alla procedura di servizio LEGGINUMERO.
PULISCITESTO ripulisce la parte dello schermo dedicata al testo e vi colloca il cursore in cima.
ACASO ha come input un numero, e fornisce in output un numero scelto a caso fra 0 e il numero di input diminuito di uno. Ogni volta, ACASO fornisce numeri sempre diversi. Così, ACASO 5 sforna numeri presi a caso fra 0 e 4, ACASO 99 numeri presi a caso fra 0 e 98, e così via.
SCRIVI stampa una parola, o una lista, senza spostare il cursore all'inizio della linea successiva (come fa invece il comando STAMPA). LEGGINUMERO attende che venga battuto un numero alla tastiera e poi lo passa in output: funziona come LEGGIRIG (che è un comando originale del Logo), ma invece di dare l'output in forma di lista, come fa questa, lo dà in forma di numero. Il suo funzionamento è descritto nel Capitolo 14 "Funzionamento delle procedure di servizio". Per ora, si può benissimo usare LEGGINUMERO come se fosse un comando originale (si dice anche: un comando nativo) del Logo. LEGGINUMERO torna utile in molte occasioni.
Vediamo ora come funziona l'insieme:

```
PER INDOVINUMERO
PRESENTO
ESTRAGGO
PRENDO
FINE
```

La INDOVINUMERO è una superprocedura, che si limita a richiamare altre tre procedure, che fanno tutto il lavoro.
Passiamo alla PRESENTO:

```
PER PRESENTO
PULISCITESTO
STAMPA [STO PENSANDO A UN NUMERO FRA 0 E 100!]
STAMPA [INDOVINA QUAL E'?]
FINE
```

Con la prima istruzione si ripulisce lo schermo e con le altre due si visualizza il messaggio di presentazione del gioco.
Passiamo alla ESTRAGGO:

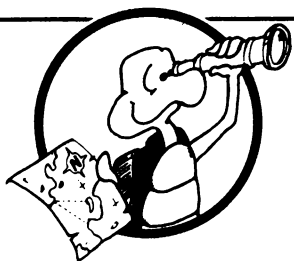
```
PER ESTRAGGO
ASSEGNA "NUMERO 1+ACASO 99
FINE
```

Con ESTRAGGO viene creata una nuova variabile chiamata "NUMERO e le si assegna un valore scelto a caso, compreso fra 1 e 99 (aggiungendo 1 all'output di ACASO 99, che ha un output casuale compreso fra 0 e 98). Ora c'è la PRENDO:

```
PER PRENDO
SCRIVI ">
ASSEGNA "TENTATIVO LEGGINUMERO
CONFRONTO :TENTATIVO :NUMERO
FINE
```

Come si vede, PRENDO richiama due procedure, LEGGINUMERO e CONFRONTO. Con la prima istruzione, fa comparire sullo schermo il simbolo > e con la seconda crea la variabile "TENTATIVO, assegnandole come valore il numero battuto alla tastiera (è la LEGGINUMERO che lo fornisce). La terza istruzione contiene il richiamo alla CONFRONTO, con i suoi due dati di input :TENTATIVO (il numero appena battuto da chi gioca) e :NUMERO (il numero segreto, scelto a caso dal computer).
Ecco CONFRONTO:

```
PER CONFRONTO :TENTATIVO :NUMERO
SE :TENTATIVO = :NUMERO STAMPA [INDOVINATO!] STOP
SE :TENTATIVO> :NUMERO STAMPA [NO: CALA] PRENDO STOP
SE :TENTATIVO< :NUMERO STAMPA [NO: AUMENTA]
   PRENDO STOP
FINE
```

La prima istruzione verifica se il numero tentato è quello giusto: se lo è, la procedura si arresta e il gioco è finito. La seconda istruzione controlla se il numero tentato è più grande del numero segreto: se lo è fa comparire la scritta NO: CALA e richiama una nuova PRENDO per il nuovo tentativo di chi gioca. La terza istruzione invece controlla se il numero tentato è più piccolo del numero segreto: in questo caso fa comparire la scritta NO: AUMENTA e richiama anch'essa una nuova PRENDO.
La PRENDO, quando ha il nuovo tentativo del giocatore, richiama a sua volta una nuova CONFRONTO, che verifica il nuovo tentativo. CONFRONTO e PRENDO richiamano l'una sempre nuovi replicanti dell'altra, finché il numero segreto non è indovinato.
Se il giocatore è bravo (o fortunato) il gioco è breve, se no può durare anche a lungo.

ESPLORAZIONE

Si può migliorare la INDOVINUMERO, magari facendole tenere il conto di quanti tentativi occorrono per indovinare il numero segreto. Ci vuole una nuova variabile, che si può chiamare "CONTO, da aumentare di uno ogni volta che viene richiamata la PRENDO (visualizzando magari anche il suo valore). Ecco qui come potrebbe essere una giocata con questa modifica:

```
INDOVIQUANTO
STO PENSANDO AD UN NUMERO FRA 0 E 100!
INDOVINA QUAL E'?
1>50
NO: CALA
2>25
NO: AUMENTA
3>35
NO: CALA
4>30
NO: CALA
5>28
INDOVINATO!
```

Per fare la modifica, basta aggiungere tre istruzioni alle procedure già scritte. La prima istruzione va in INDOVINUMERO (che diventa poi INDOVIQUANTO, per distinguerla dalla prima) e crea la nuova variabile, assegnando il valore zero di partenza:

```
ASSEGNA "CONTO 0
```

Le altre due vanno in PRENDO:

```
ASSEGNA "CONTO :CONTO+1
SCRIVI :CONTO
```

Vale la pena di provare.

9.6 MATQUIZ

Il Logo è bravissimo a fare i conti! Perché non ci alleniamo con lui? Possiamo giocare a MATQUIZ che, nella sua prima versione, va bene per l'addizione (se si vogliono anche le altre operazioni, basta fare la modifica).
Il programma è già pronto: basta copiare le istruzioni riportate più avanti in questo stesso paragrafo; occorre anche la procedura di servizio LEGGINUMERO, che si trova nell'Appendice A.
Ecco una giocata:

```
MATQUIZ
QUANTE DOMANDE VUOI PROVARE?
2
DOMANDA 1
35+41 = 76
GIUSTO
PER FAVORE, BATTI IL TASTO RETURN
DOMANDA 2
78+43 = 111
NO: LA RISPOSTA E' 121
PER FAVORE, BATTI IL TASTO RETURN
PUNTEGGIO OTTENUTO: 1
SU 2 DOMANDE
```

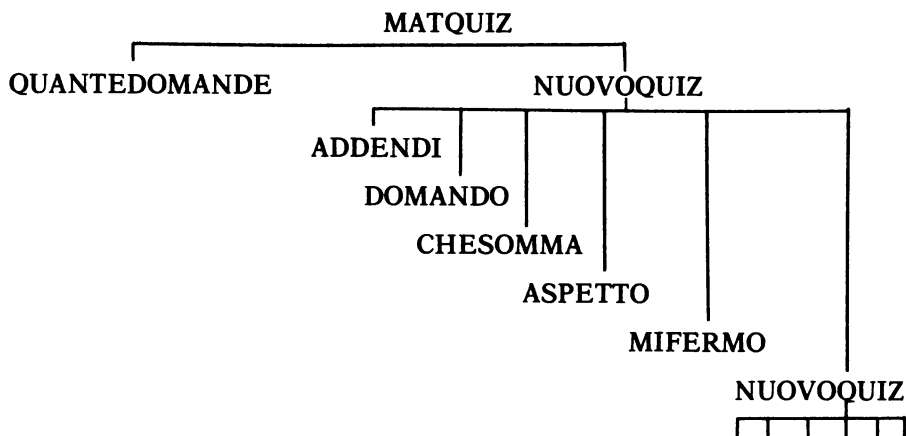
Vediamo le funzioni principali del programma:

1. Vedere quante domande si vogliono.
2. Scegliere due numeri e presentare la prima domanda.
3. Vedere se la risposta battuta alla tastiera è corretta (se lo è, scrivere "GIUSTO" e aumentare il punteggio di uno).
4. Vedere se sono state fatte tutte le domande richieste, per fermarsi, scrivendo il punteggio ottenuto, oppure proseguire con un altro caso (aumentando di uno il contatore delle domande fatte).

Ecco le procedure:

```
PER MATQUIZ
PULISCITESTO
QUANTEDOMANDE
ASSEGNA "CONTO 1
ASSEGNA "PUNTI 0
NUOVOQUIZ :CONTO :QUANTE :PUNTI
FINE
```

```
PER QUANTEDOMANDE
STAMPA [QUANTE DOMANDE VUOI PROVARE?]
ASSEGNA "QUANTE LEGGINUMERO
FINE
PER NUOVOQUIZ :CONTO :QUANTE :PUNTI
PULISCITESTO
ADDENDI
DOMANDO
CHESOMMA
ASPETTO
SE :CONTO = :QUANTE MIFERMO STOP
NUOVOQUIZ (:CONTO+1) :QUANTE :PUNTI
FINE
PER ADDENDI
ASSEGNA "A1 ACASO 100
ASSEGNA "A2 ACASO 100
ASSEGNA "SOMGIUSTA :A1+:A2
FINE
PER DOMANDO
STAMPA FRASE [DOMANDA] :CONTO
STAMPA [ ]
SCRIVI (FRASE: A1 [+] :A2 [=])
FINE
PER CHESOMMA
ASSEGNA "SOMMA LEGGINUMERO
VERIFICA :SOMMA = :SOMGIUSTA
SEVERO STAMPA [GIUSTO]
SEVERO ASSEGNA "PUNTI :PUNTI+1
SEFALSO STAMPA FRASE [NO: LA RISPOSTA E']: SOMGIUSTA
FINE
PER ASPETTO
STAMPA [PER FAVORE, BATTI IL TASTO RETURN]
STAMPA LEGGIRIG
FINE
PER MIFERMO
PULISCITESTO
STAMPA FRASE [PUNTEGGIO OTTENUTO:] :PUNTI
STAMPA (FRASE [SU] :QUANTE [DOMANDE])
FINE
```

Anche in questo caso conviene tracciare lo schema di connessione delle procedure:

Vediamo ora il comportamento delle procedure, una alla volta. Ecco le prime due:

```
PER MATQUIZ
PULISCITESTO
QUANTEDOMANDE
ASSEGNA "CONTO 1
ASSEGNA "PUNTI 0
NUOVOQUIZ :CONTO :QUANTE :PUNTI
FINE
PER QUANTEDOMANDE
STAMPA [QUANTE DOMANDE VUOI PROVARE?]
ASSEGNA "QUANTE LEGGINUMERO
FINE
```

Come si vede, MATQUIZ ripulisce lo schermo, richiama QUANTEDOMANDE (che si fa indicare da chi gioca il numero di domande voluto) e fissa il valore iniziale del punteggio (a zero) e del contatore delle domande fatte (a uno); fatto ciò, richiama la NUOVOQUIZ.
Ecco NUOVOQUIZ:

```
PER NUOVOQUIZ :CONTO :QUANTE :PUNTI
PULISCITESTO
ADDENDI
DOMANDO
CHESOMMA
ASPETTO
SE :CONTO = :QUANTE MIFERMO STOP
NUOVOQUIZ (:CONTO+1) :QUANTE :PUNTI
FINE
```

All'inizio, :CONTO vale 1, :QUANTE vale il numero di domande chiesto dal giocatore (e non varia nel corso del gioco), mentre :PUNTI vale 0. Dopo aver ripulito lo schermo, la NUOVOQUIZ chiama per prima la ADDENDI:

```
PER ADDENDI
ASSEGNA "A1 ACASO 100
ASSEGNA "A2 ACASO 100
ASSEGNA "SOMGIUSTA :A1+:A2
FINE
```

Dopo aver estratto a caso i due addendi (fra 0 e 100), la ADDENDI ne calcola la somma, dando al valore trovato il nome "SOMGIUSTA, e poi chiama la DOMANDO:

```
PER DOMANDO
STAMPA FRASE [DOMANDA] :CONTO
STAMPA [ ]
SCRIVI (FRASE :A1 [+] :A2 [=])
FINE
```

La DOMANDO si limita a visualizzare gli addendi (visto come si fa a non saltare la riga, dopo il segno = ?) e chiama subito la CHESOMMA:

```
PER CHESOMMA
ASSEGNA "SOMMA LEGGINUMERO
VERIFICA :SOMMA = :SOMGIUSTA
SEVERO STAMPA [GIUSTO]
SEVERO ASSEGNA "PUNTI :PUNTI+1
SEFALSO STAMPA FRASE [NO: LA RISPOSTA E']: SOMGIUSTA
FINE
```

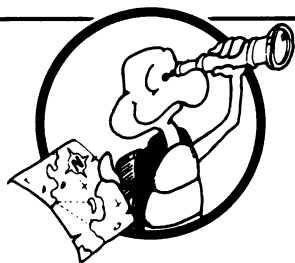
Come si vede, CHESOMMA prende la risposta battuta dal giocatore e controlla se corrisponde alla somma giusta (con i comandi VERIFICA, SEVERO e SEFALSO). Se la risposta è giusta, aumenta di uno il punteggio. I comandi VERIFICA, SEVERO e SEFALSO sono simili al comando SE. Il dato di input di VERIFICA è una relazione, che può essere o vera o falsa; se la relazione è vera (cioè vale la parola "VERO) vengono eseguiti i comandi che seguono SEVERO, mentre nel caso contrario vengono eseguiti i comandi che seguono SEFALSO (nella stessa istruzione). La NUOVOQUIZ prosegue chiamando la ASPETTO:

```
PER ASPETTO
STAMPA [PER FAVORE, BATTI IL TASTO RETURN]
```

```
STAMPA LEGGIRIG
FINE
```

Con la ASPETTO, è il giocatore che decide quando passare alla domanda successiva. La LEGGIRIG aspetta fino a quando non viene battuto almeno un tasto (qualunque tasto, seguito dal tasto RETURN). Il comando STAMPA serve solo per saltare una riga (se il giocatore segue le istruzioni, batte solo il tasto RETURN, e l'output di LEGGIRIG è la lista vuota [ ]). La ASPETTO torna utile in molti altri casi, quando si tratta di lasciar decidere all'utilizzatore il momento di proseguire.
L'ultimo controllo di NUOVOQUIZ riguarda il numero di domande fatte: quando raggiunge il numero di domande richiesto, viene chiamata la MIFERMO, che stampa il punteggio, e la procedura si arresta.
Ecco la MIFERMO:

```
PER MIFERMO
PULISCITESTO
STAMPA FRASE [PUNTEGGIO OTTENUTO:] :PUNTI
STAMPA (FRASE [SU] :QUANTE [DOMANDE])
FINE
```

Se il numero di domande fatte non è pari al numero totale stabilito dal giocatore, la NUOVOQUIZ richiama un'altra NUOVOQUIZ, che si trova il proprio :CONTO aumentato di uno, mentre :QUANTE vale sempre lo stesso, e :PUNTI è aumentato di uno solamente se la risposta data era corretta.

ESPLORAZIONE

Ecco qui un elenco di possibili modifiche, che è interessante provare:

- Lasciare decidere a chi gioca quanto grandi possono essere gli addendi. Bisogna allora dare come input di ACASO (in ADDENDI) il numero battuto dal giocatore, invece del numero 100 del nostro esempio.
- Disporre gli addendi in modo diverso. Ad esempio:

```
DOMANDA 1
    17
    +
    28
   ———
```

Si possono stampare spazi vuoti usando il simbolo '.
Provate questo esempio:

```
STAMPA FRASE "' '25
```

(Gli spazi fra i segni ' vengono stampati tali e quali).

- Per fare qualcosa di estroso, costruire i numeri e i simboli matematici in modo grafico (con la tartaruga) e visualizzare le domande e le risposte sullo schermo grafico. Questo è già un progetto grafico impegnativo.
- Fare in modo che le domande diventino più difficili, se il giocatore risponde giusto (si può ad esempio rendere sempre più grande il massimo numero estraibile a caso).
- Cambiare i messaggi scritti dal computer, man mano che il gioco prosegue (si possono cambiare ad ogni nuova domanda).
- Fornire suggerimenti, se la risposta è sbagliata (questo non è facile perché bisogna prima conoscere quali possono essere i suggerimenti utili, nelle varie situazioni).
- Estendere il quiz alle altre operazioni (sottrazione, moltiplicazione e divisione). Si può anche preparare una superprocedura che fa scegliere a chi gioca il tipo di operazione in cui si vuole allenare.

SUGGERIMENTO

È molto difficile realizzare un programma interattivo facile da usare e che aiuti chi lo usa. In effetti, proprio qui sta la debolezza di tanti programmi di cosiddetta CAI (Computer Aided Instruction), a tal punto che in molti casi si ottiene solo una gran perdita di tempo.
D'altro canto, è vero che proprio affrontando questo problema, non facile, un programmatore ha l'occasione di accrescere la propria esperienza e di affinare la propria tecnica.

Inoltre, i programmi scritti in questo modo sono in genere una fonte continua di ispirazione per i nuovi casi applicativi che man mano si presentano. Si può tentare un progetto molto ambizioso, con un gruppo di bravi allievi, facendo realizzare a ciascuno un piccolo programma di CAI, che viene poi provato da tutti gli altri. Si ottiene così, se tutto va bene, un buon esercizio di programmazione e un'ottima prova di collaborazione nell'apprendimento.

10

TIRO.A.SEGNO

Abbiamo visto come si gioca a TIRO.A.SEGNO nel Capitolo 3: si tratta di puntare la tartaruga su un bersaglio che compare sullo schermo e di "spararla", per vedere se lo si colpisce. È una buona occasione per allenare un principiante alla manovra della tartaruga e alla stima di angoli e distanze. Prima di proseguire conviene rinfrescare la memoria del gioco, riprendendo il Capitolo 3.
Qui si descrive come funzionano le diverse procedure che formano TIRO.A.SEGNO, e si indica qualche spunto per estenderlo e migliorarlo. Più in generale, in questo capitolo e nei tre che lo seguono si affronta l'argomento della realizzazione di programmi interattivi, quelli cioè che per funzionare si basano sull'interazione fra il programma stesso e la persona che lo usa.
In pratica, un programma interattivo fa fare al computer cose che dipendono, almeno in parte, da ciò che fa l'utilizzatore (e viceversa, l'utilizzatore compie azioni che sono in parte condizionate dal computer). Un primo esempio di programma interattivo è già comparso nel Capitolo 9 (i giochi a quiz...): qui si riprendono le idee di quel capitolo (e di altri ancora) per realizzare un gioco interattivo con la tartaruga.
Il metodo seguito in questo capitolo (e in quelli che lo seguono) consiste nel partire da un programma già pronto, anche se un po' grezzo: per prima cosa si spiegano le procedure esistenti, poi si indicano le modifiche e le estensioni che migliorano il programma e lo rendono più interessante da usare. Si trova così anche l'occasione di approfondire la conoscenza del Logo e di capire meglio il suo funzionamento, in vista della realizzazione di progetti totalmente nuovi. Alla fine del Capitolo 13, il lettore avrà un sacco di idee da mettere in pratica, e lo potrà fare in modo auto-

nomo perché conoscerà il Logo a sufficienza.
Tutte le procedure presentate (comprese le procedure di servizio che le fanno funzionare) si trovano nell'Appendice A.

Comando	Abbreviazione	Esempi
VAXY		VAXY 100 30, VAXY 10 (−20)
VAX		VAX 50, VAX (−100)
VAY		VAY 35, VAY (−10)
ASDIR		ASDIR 90, ASDIR 30
SEME		ASDIR 90, ASDIR 30

Vengono usati i seguenti file "TIRO.A.SEGNO, "DISTANZA, "CERCHIETTO, "LEGGINUMERO, "ASPETTA dell'Appendice A.

Vengono usate nuove procedure di servizio:

Nome	Esempio
DISTANZA	STAMPA DISTANZA 0 100
	SE (DISTANZA :XPART :YPART< 10) STOP
CERCHIETTO	CERCHIETTO 50
ASPETTA	ASPETTA 100

SUGGERIMENTO

Per chi impara il Logo, l'esperienza di costruire un programma del tutto nuovo vale quanto quella di imparare un programma già fatto da altri, modificandolo magari in un secondo tempo. Del resto, anche in pratica accade molto spesso di realizzare un programma partendo da programmi realizzati da altri, che sono poi modificati e adattati alle esigenze del momento.
I giochi come TIRO.A.SEGNO sono una delle applicazioni più comuni del computer: in effetti, per molti principianti sembra proprio che il maggiore interesse si concentri proprio sull'interazione con il computer (domanda e risposta, manovra di un oggetto, ecc.); anzi, è proprio in questo campo che chi impara trova la prima occasione di affinamento e di progresso.
L'elaborazione dati (come si dice) è tutta qua, in nuce: creazione, memorizzazione, manipolazione di informazioni con il computer. Sono proprio questi gli argomenti di fondo sottesi dai progetti presentati in questo capitolo e

in quelli che lo seguono. Più in generale, dalla loro lettura si dovrebbe ricavare un'idea di cos'è l'elaborazione dati, di come si realizza e del perché è importante.
Come ho già suggerito nel Capitolo 3, il TIRO.A.SEGNO è un'eccellente occasione di collaborazione fra un giovane principiante e una persona più esperta (dai dieci anni in su, magari anche un adulto): i due potrebbero dedicarsi in coppia alla modifica del gioco, così da avere entrambi un interesse e un entusiasmo maggiori.
In una scuola si potrebbe addirittura far collaborare una delle prime classi con una delle ultime: un'ottima opportunità per lo scambio di idee e per lo stimolo di proposte interessanti. Certo, sarebbe meglio evitare di ridurre il tutto a una gara per avere il gioco migliore di quello degli altri, e saper cogliere l'occasione per un confronto delle idee e delle esperienze.

10.1 Nuovi comandi Logo e nuove procedure di servizio in TIRO.A.SEGNO

Finora abbiamo visto i comandi elementari di manovra della tartaruga: AVANTI, INDIETRO, DESTRA e SINISTRA. Questi comandi consentono di spostare la tartaruga ovunque, da qualunque punto di partenza. Esiste anche, nel Logo, un modo diverso di manovrare la tartaruga, che si basa sull'indicazione delle coordinate X e Y del punto di arrivo.
Le coordinate X e Y (dette anche coordinate cartesiane) sono misurate dal centro dello schermo, come si vede in Figura 10.1.
In TIRO.A.SEGNO, si fa uso delle coordinate X e Y per indicare la posizione del bersaglio e quella della tartaruga all'inizio del gioco. Inoltre, la

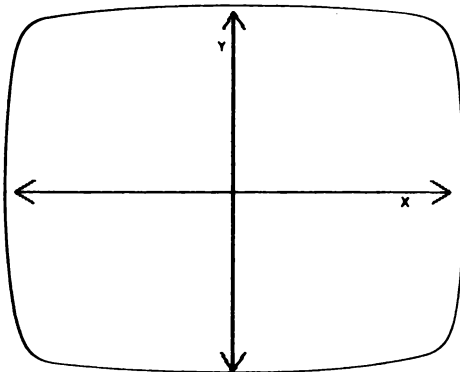



Figura 10.1 Le coordinate X e Y di un punto dello schermo indicano la distanza di questo dal centro, in orizzontale e in verticale

tartaruga può essere spostata direttamente in qualunque punto dello schermo, con il comando VAXY, che richiede due dati di input: la coordinata X (orizzontale) e la coordinata Y (verticale) del nuovo punto. Valori positivi di X spostano la tartaruga nella metà destra dello schermo (valori negativi, nella metà sinistra). Valori positivi di Y spostano la tartaruga nella metà superiore dello schermo (e valori negativi, invece, nella metà inferiore).
Provate questi esempi:

```
DISEGNA
VAXY 20 30
VAXY (-50) 75
VAXY (-100) (-20)
VAXY 45 (-80)
```

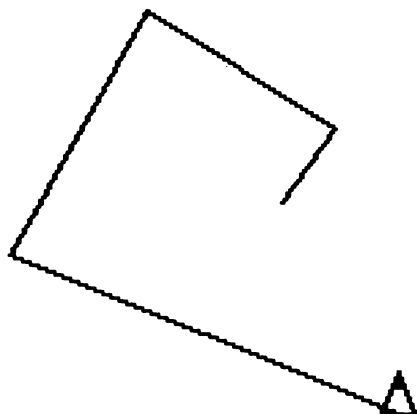



Figura 10.2 Il comando VAXY sposta la tartaruga direttamente nel punto indicato, senza alterarne l'orientamento

Attenzione quando si fornisce un valore negativo (come −50) ad un comando: è sempre meglio racchiuderlo fra parentesi. Possono sorgere infatti malintesi. Provate questo:

```
VAXY 100 -50
```

Il Logo, in questo caso, protesta perché esegue la sottrazione, invece di lasciare separati i due numeri, e poi si trova con un solo dato per VAXY, invece dei due che sono necessari al comando. Si deve battere invece:

```
VAXY 100 (-50)
```

Come si vede in Figura 10.2, il comando VAXY sposta la tartaruga senza modificarne l'orientamento: questo viene invece cambiato dal comando ASDIR. Provate questi casi:

```
ASDIR 45
ASDIR 300
ASDIR 400
ASDIR 1000
```

Figura 10.3 Il comando ASDIR assegna un nuovo orientamento alla tartaruga

Se si fornisce a ASDIR un valore di input maggiore di 360, questo viene immediatamente modificato sottraendo 360. Così, con il comando ASDIR 400 si ottiene lo stesso effetto di ASDIR 40 (infatti 400 −360 è pari a 40) e ASDIR 1000 vale come ASDIR 280 (infatti 280 è pari a 1000 −360 −360).
I comandi VAX e VAY richiedono un solo dato di input, e modificano solo la coordinata X (o Y, rispettivamente), lasciando l'altra coordinata invariata (in pratica, così la tartaruga si muove solo in orizzontale, o in verticale).
In TIRO.A.SEGNO sono anche usate le procedure di servizio DISTANZA, CERCHIETTO, LEGGINUMERO e ASPETTA (sono riportate nell'Appendice A).
La procedura DISTANZA lavora su due dati di input, che sono le coordinate X e Y di un punto: essa calcola la distanza fra tale punto e la tartaruga (nel TIRO.A.SEGNO, serve per trovare quanto è distante la tartaruga dal centro del bersaglio). Passiamo ora a qualche esempio (prima bisogna caricare dal disco DISTANZA, o averla ricopiata dall'Appendice A):

```
DISEGNA
STAMPA DISTANZA 100 0
SE (DISTANZA 100 0< 10) STAMPA [CENTRO!]
SE (DISTANZA 5 5< 10) STAMPA [CENTRO!]
```

Nel primo caso, il computer stampa il valore della distanza fra il punto in cui si trova la tartaruga e il punto di coordinate X=100 e Y=0. Negli altri due casi, esso verifica se la tartaruga è più vicina di 10 passi da un punto dato (che può essere il centro di un bersaglio), per stampare una scritta in caso di successo.
La procedura CERCHIETTO ha un solo dato di input, che indica il raggio

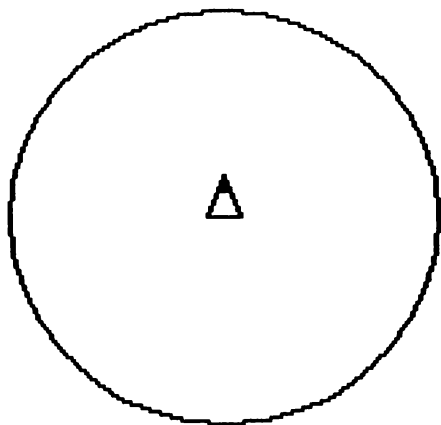



Figura 10.4 Con CERCHIETTO si traccia un cerchio intorno al punto in cui si trova la tartaruga

del cerchio, tracciato intorno al punto in cui si trova al momento la tartaruga (anche qui, prima di provarla bisogna caricarla da un disco o ricopiarla dall'Appendice A):

CERCHIETTO 50

La CERCHIETTO serve in TIRO.A.SEGNO, unita a VAXY, per spostare la tartaruga in un certo punto e tracciarvi intorno un cerchietto che rappresenta il bersaglio: questo stesso punto serve poi da input a DISTANZA, per controllare se la tartaruga ha colpito il bersaglio.
La LEGGINUMERO, già vista nel Capitolo 9, è simile al comando LEGGIRIG, nativo del Logo. Essa rimane in attesa che venga battuto un numero alla tastiera, e lo passa in output (la LEGGIRIG produce invece come output una lista, non un numero).
Dopo aver caricato la LEGGINUMERO dal disco o averla ricopiata dall'Appendice A, si può provare un esempio:

STAMPA LEGGINUMERO+55

Dopo questo comando, il computer se ne sta in attesa che venga battuto un numero (seguito dal tasto RETURN), poi somma a questo 55 e stampa il risultato. Attenzione, se si batte qualcosa che non è un numero, il Logo protesta.
La procedura ASPETTA fa attendere il computer per un po': quanto? Dipende dal valore di input (con 100, il computer rimane in attesa per circa due secondi). Ecco qualche esempio:

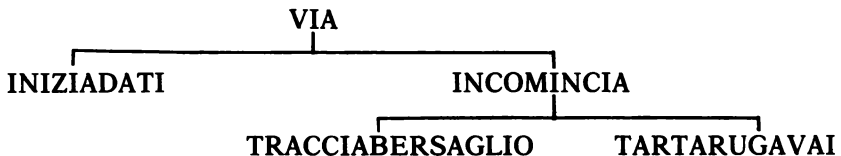
```
ASPETTA 100
ASPETTA 200
ASPETTA 1000
```

Dopo aver visto VAXY, ASDIR, DISTANZA, CERCHIETTO, LEGGINUMERO e ASPETTA siamo pronti a capire come funziona TIRO.A.SEGNO (ci sono altri comandi e procedure di servizio di minore importanza, che verranno spiegati man mano che li si incontra).

10.2 TIRO.A.SEGNO in dettaglio

TIRO.A.SEGNO si compone di otto procedure. Due di esse, VIA e TIRO, sono superprocedure, cioè comandi che mettono in moto tutta una serie di azioni. Le procedure secondarie (in pratica, quelle che svolgono la maggior parte del lavoro) sono INIZIADATI, INCOMINCIA, TRACCIABERSAGLIO, TARTARUGAVAI, COLPITO e MANCATO. Compaiono anche quattro procedure di servizio, già viste prima, DISTANZA, ASPETTA, CERCHIETTO e LEGGINUMERO.
A questo punto, si carica dal disco il file "TIRO.A.SEGNO, oppure si ricopiano man mano le procedure dall'Appendice A (comprese le procedure di servizio).

LA SUPERPROCEDURA VIA

Incominciamo da VIA e dalle procedure che essa richiama: INIZIADATI, INCOMINCIA, TRACCIABERSAGLIO e TARTARUGAVAI. Ecco qui il diagramma di connessione:

Come si vede, VIA controlla due procedure, INIZIADATI e INCOMINCIA: quest'ultima, a sua volta, ne controlla altre due, TRACCIABERSAGLIO e TARTARUGAVAI. Ecco le istruzioni:

```
PER VIA
INIZIADATI
INCOMINCIA
FINE
```

Con INIZIADATI, la VIA fissa la situazione di partenza; poi chiama la IN-COMINCIA che disegna il bersaglio e colloca la tartaruga nella posizione di partenza.
Ecco INIZIADATI:

```
PER INIZIADATI
SEME
ASSEGNA "COLPI 0
ASSEGNA "XBERS (90-10*ACASO 19)
ASSEGNA "YBERS (80-10*ACASO 6)
ASSEGNA "XPART (90-10*ACASO 19)
ASSEGNA "YPART (-10 *ACASO 3)
ASSEGNA "OPART (10*ACASO 36)
FINE
```

Con INIZIADATI, si assegnano i nomi a tutte le variabili che riguardano i dati necessari allo svolgimento del gioco. Il valore iniziale viene assegnato con il comando ASSEGNA.
Ad esempio, il comando:

```
ASSEGNA "COLPI 0
```

crea una variabile chiamata "COLPI, che parte col valore zero.
Nelle cinque righe successive, si impiega il comando ACASO (nativo del Logo) per generare i valori iniziali delle coordinate del bersaglio e della tartaruga, e dell'orientamento di questa. Con ACASO, il gioco riprende ogni volta con una situazione diversa.
"XBERS e "YBERS sono i nomi per le coordinate X e Y del centro del bersaglio. Ad ogni ripresa del gioco, la coordinata X del bersaglio si trova a caso fra −90 e 90, mentre la coordinata Y cade a caso fra 30 e 80.
"XPART, "YPART e "OPART nominano le coordinate X e Y e l'orientamento di partenza della tartaruga. Questa parte con coordinata X compresa fra −90 e 90, coordinata Y compresa fra −20 e 0, e orientamento compreso fra 0 e 350 gradi. Il primo comando, SEME, fa in modo che il computer scelga numeri a caso sempre diversi, ogni volta che si gioca.
Ed ecco INCOMINCIA:

```
PER INCOMINCIA
DISEGNA
NASTARTA
TRACCIABERSAGLIO :XBERS :YBERS
TARTARUGAVAI :XPART :YPART :OPART
MOSTARTA
FINE
```

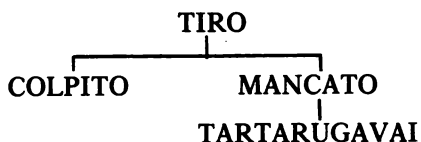
Con TRACCIABERSAGLIO e TARTARUGAVAI, si disegna il bersaglio e la tartaruga, nella posizione di partenza fornita da INIZIADATI. Ecco le istruzioni:

```
PER TRACCIABERSAGLIO :XBERS :YBERS
SULAPENNA
VAXY :XBERS :YBERS
CERCHIETTO 10
FINE
```

Come si vede, con il comando VAXY e la procedura di servizio CERCHIETTO viene disegnato un bersaglio circolare intorno al punto dato da INIZIADATI. Ecco invece le istruzioni di TARTARUGAVAI, che è simile a TRACCIABERSAGLIO (non traccia il cerchietto, però considera anche l'orientamento della tartaruga, pure determinato dalla INIZIADATI):

```
PER TARTARUGAVAI :XPART :YPART :OPART
SULAPENNA
VAXY :XPART :YPART
ASDIR :OPART
FINE
```

LA SUPERPROCEDURA TIRO

TIRO è la seconda superprocedura che regola il gioco, ed entra in azione quando il giocatore ha ormai puntato la tartaruga sul bersaglio ed è pronto a tirare il colpo. Come si vede dal diagramma di connessione riportato qui sotto, vengono chiamate tre procedure secondarie (oltre alle procedure di servizio DISTANZA, LEGGINUMERO e ASPETTA, che non sono indicate nel diagramma):

La TIRO è il cuore del gioco. Vediamo allora per bene le sue istruzioni:

```
PER TIRO
ASSEGNA "COLPI = :COLPI+1
STAMPA [A CHE DISTANZA?]
ASSEGNA "GITTATA LEGGINUMERO
```

```
GIULAPENNA AVANTI :GITTATA
VERIFICA DISTANZA :XBERS :YBERS< 10
SEVERO COLPITO
SEFALSO MANCATO
FINE
```

Nell'istruzione:

```
ASSEGNA "COLPI = :COLPI+1
```

compare la variabile "COLPI, che tiene conto del numero di colpi tirati. Ogni volta che si batte TIRO, questo contatore viene aumentato di uno. Il comando ASSEGNA ha due dati di input: il primo è "COLPI, che crea una nuova variabile con questo nome; il secondo è :COLPI+1, che indica il valore della nuova variabile (qui è il valore della vecchia variabile aumentato di uno). Questo modo di fare è molto comune: si parla, in gergo, di incremento del valore di "COLPI, o anche di aggiornamento della variabile "COLPI.
La seconda istruzione è semplice:

```
STAMPA [A CHE DISTANZA?]
```

chiede al giocatore di indicare la distanza del nuovo tiro.
La terza istruzione:

```
ASSEGNA "GITTATA LEGGINUMERO
```

aspetta che il giocatore batta un numero e lo assegna alla nuova variabile chiamata "GITTATA.
La quarta istruzione:

```
GIULAPENNA AVANTI :GITTATA
```

sposta la tartaruga in avanti, per il numero di passi indicato da :GITTATA (e lascia la traccia del percorso sullo schermo).
Passiamo alla quinta:

```
VERIFICA (DISTANZA :XERS :YBERS)< 10
```

qui si controlla se la distanza della tartaruga dal centro del bersaglio è inferiore a 10 passi. Il risultato può essere o la parola "VERO oppure la parola "FALSO, e viene memorizzato dal computer, almeno temporaneamente. Le parentesi che racchiudono il comando DISTANZA e i suoi dati di input non sono necessarie al Logo: rendono però più facile la lettura visiva dell'istruzione.

La sesta istruzione:

```
SEVERO COLPITO
```

agisce se il risultato di VERIFICA è la parola "VERO, per chiamare la procedura COLPITO. Se invece l'output di VERIFICA è la parola "FALSO, agisce la settima istruzione:

```
SEFALSO MANCATO
```

viene allora chiamata la procedura MANCATO, che aspetta un momento e poi rimanda la tartaruga al punto di partenza.
Ecco COLPITO:

```
PER COLPITO
STAMPA [CONGRATULAZIONI! BERSAGLIO COLPITO!]
STAMPA (FRASE [CI SONO VOLUTI SOLO] :COLPI [COLPI])
FINE
```

Attenzione, le parentesi nella seconda istruzione sono necessarie, perché il comando FRASE ha qui più di due dati di input.
Ecco MANCATO:

```
PER MANCATO
STAMPA FRASE [MANCATO! COLPI TIRATI:] :COLPI
ASPETTA 200
TARTARUGAVAI :XPART :YPART :OPART
FINE
```

Dopo aver comunicato che il bersaglio è mancato, e quanti colpi sono già stati tirati, il computer aspetta un pochino e poi richiama la TARTARUGAVAI (quella usata all'inizio del gioco) che ricolloca la tartaruga al punto di partenza, pronta a ricominciare).

SUGGERIMENTO

Il programma TIRO.A.SEGNO fa uso di variabili globali (o, come si è detto prima per amor di semplicità, pubbliche). Questo tipo di variabile, che si crea con il comando ASSEGNA, è diverso dal tipo usato finora, cioè la variabile cosiddetta "privata", che vale solo all'interno della procedura alla quale appartiene (in gergo informatico, si parla di variabili locali).
Con le variabili globali si realizza in pratica una specie di comunicazione fra tutte le procedure che le usano, perché l'aggiornamento del valore di una variabile globale, fatto in una procedura, vale anche per tutte le altre. Quando si impiegano variabili globali, bisogna evitare assolutamente ogni duplicazione di nomi: questo non è necessario invece con le variabili locali, proprio perché esse sono limitate alla procedura con cui sono create. È perciò possibile avere due procedure che usano gli stessi nomi per le proprie variabili locali, che rimangono sempre distinte (molti programmatori preferiscono le variabili locali proprio per questo motivo).

10.3 Miglioriamo TIRO.A.SEGNO

Chiunque, dopo aver giocato per un po' a TIRO.A.SEGNO, si fa una propria idea delle modifiche che renderebbero il gioco più interessante. E proprio qui viene il bello, perché TIRO.A.SEGNO, essendo scritto in Logo, può essere modificato da chi lo usa, a differenza di tanti altri giochi da computer. Fra le tante possibili modifiche, ecco le più immediate:

1. Aumentare la vivacità del gioco, con un effetto speciale ogni volta che il bersaglio è colpito.
2. Rendere il gioco più impegnativo (con un bersaglio ridotto, o magari evitando di tracciare la traiettoria della tartaruga, così da ostacolare la stima dell'errore di mira).
3. Rendere il gioco più facile, ingrandendo il bersaglio o facendo dare dal computer consigli per una mira migliore.
4. Arricchire i messaggi del computer, con battute di spirito o magari con istruzioni d'uso per chi non conosce ancora il gioco.
5. Preparare un SUPERTIRO.A.SEGNO che combini alcuni dei miglioramenti descritti e proponga la scelta del livello di gioco (facile, intermedio, difficile).

Vedremo ora come eseguire le modifiche (naturalmente, così facendo si impara a modificare qualunque procedura in Logo).

10.4 Per un gioco più vivace

Si può ravvivare il gioco con l'inserimento di un fatto nuovo quando il bersaglio è colpito: che ne dite di una bella esplosione? Ecco qua:

```
PER BANG :GRANDE
NASTARTA
RIPETI 18[AVANTI :GRANDE INDIETRO :GRANDE DESTRA 20]
FINE
```

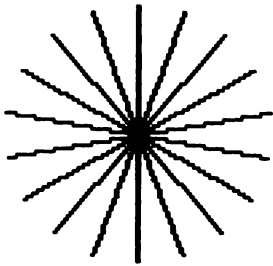



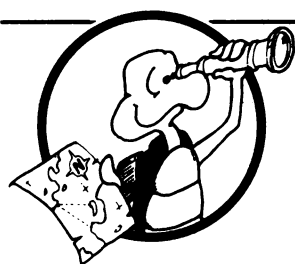
Figura 10.5 BANG! Bersaglio colpito!

Variando il dato di input di BANG, si ottiene un'esplosione più o meno grande.
Dove mettiamo la BANG? Osservando la procedura TIRO, nel paragrafo 10.2, si vede che esiste già la COLPITO, che agisce in caso di successo, e che scrive un certo messaggio. La nuova BANG può essere inserita nella TIRO, richiamandola subito dopo la COLPITO, oppure può essere inserita in quest'ultima. Nel primo caso, la sesta istruzione di TIRO diventa:

```
SEVERO COLPITO BANG 20
```

Nel secondo caso, l'inserimento in COLPITO può precedere o seguire la scritta:

```
PER COLPITO
STAMPA [CONGRATULAZIONI! BERSAGLIO COLPITO!]
STAMPA (FRASE [CI SONO VOLUTI SOLO] :COLPI [COLPI])
BANG 20
FINE
```

Per chi gioca, un modo vale l'altro (l'effetto è lo stesso): si può scegliere il modo che sembra più chiaro, o più facile da modificare (se viene in mente un'idea nuova).

ESPLORAZIONE

Perché non provare tante esplosioni diverse? Si possono fare grandi e piccole, e di tanti colori ... Oppure, si può fare in modo che ad ogni bersaglio colpito si veda un'esplosione diversa dalle altre. O magari si può fare l'esplosione più piccola se ci sono voluti molti colpi, e più grande se ne sono bastati pochi. Si può anche cambiare genere: ad esempio, il bersaglio può essere contrassegnato ad ogni nuovo centro, oppure può scomparire all'improvviso, dopo essere stato colpito...

10.5 Per un gioco più impegnativo

Molti trovano TIRO.A.SEGNO troppo facile, dopo un po' di esercizio. È il momento allora di aumentarne la difficoltà.
Un modo molto semplice è spostare la tartaruga senza che lasci traccia del percorso: con questa infatti il giocatore è facilitato, perché può regolare il colpo successivo con l'errore di mira appena commesso. Se però non c'è traccia della traiettoria, chi gioca deve fare uno sforzo in più, almeno per stimare l'errore o per ricordare i valori tentati nel colpo precedente.
Per eliminare la traiettoria, basta cancellare il comando GIULAPENNA nella quarta istruzione della procedura TIRO (vista nel paragrafo 10.2). Si può anche lasciare un'indicazione del colpo fatto, facendo magari tracciare alla tartaruga un segno "X", o qualcosa di simile, prima di riportarla alla posizione di partenza.
E perché non tracciare invece la traiettoria, come al solito, e cancellarla quando la tartaruga ritorna al punto di partenza? Si può fare indicando, nel percorso di ritorno, ASCOL −1 (oppure CANCEPENNA). Se si scrive ASCOL −1 non si assegna un colore, ma si fa sì che il pennino della tar-

taruga si comporti come una gomma per cancellare. L'effetto si vede con questo esempio:

```
DISEGNA
AVANTI 50
ASCOL -1
INDIETRO 50
ASCOL 1
```

Visto? La linea viene prima tracciata, e poi cancellata; per tracciarla di nuovo, bisogna prima richiamare ASCOL con un codice di colore normale (fra 0 e 15).
Attenzione però, perché il pennino della tartaruga deve essere giù, se si vuole che cancelli con ASCOL −1. Bisogna allora esaminare la MANCATO:

```
PER MANCATO
STAMPA FRASE [MANCATO! COLPI TIRATI:] :COLPI
ASPETTA 200
TARTARUGAVAI :XPART :YPART :OPART
FINE
```

Veramente, dopo aver scritto il messaggio, la MANCATO affida il lavoro alla TARTARUGAVAI:

```
PER TARTARUGAVAI :XPART :YPART :OPART
SULAPENNA
VAXY :XPART :YPART
ASDIR :OPART
FINE
```

La TARTARUGAVAI solleva il pennino della tartaruga (con SULAPENNA). Prepariamo allora una nuova procedura, simile a TARTARUGAVAI ma che non solleva il pennino chiamata RIPARTI:

```
PER RIPARTI :XPART :YPART :OPART
ASCOL -1
VAXY :XPART :YPART
ASDIR :OPART
ASCOL 1
FINE
```

Modifichiamo MANCATO:

```
PER MANCATO
STAMPA FRASE [MANCATO! COLPI TIRATI:] :COLPI
ASPETTA 200
RIPARTI :XPART :YPART :OPART
FINE
```

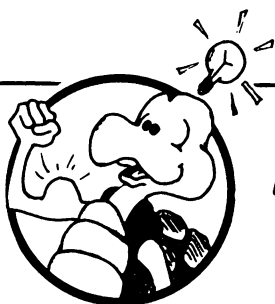
A questo punto, conviene provare che tutto funzioni come si deve. Il gioco diventa più difficile anche se si riduce il bersaglio. La modifica va fatta in TRACCIABERSAGLIO:

```
PER TRACCIABERSAGLIO :XBERS :YBERS
SULAPENNA
VAXY :XBERS :YBERS
CERCHIETTO 5
FINE
```

Come si vede, ora il bersaglio è un circoletto con 5 passi di raggio, invece di 10. Attenzione però, che va anche modificato il controllo nella quinta istruzione di TIRO:

```
VERIFICA (DISTANZA :XBERS :YBERS)< 5
```

TRABOCCHETTO

Un bug comunissimo in tutti i programmi per computer si forma quando si fa una modifica in una parte del programma e ci si dimentica di correggere tutte le altre parti che hanno riferimenti a questa.
Ad esempio, se si riduce il bersaglio e non si modifica il controllo in TIRO, si ha solo l'impressione che il bersaglio sia più piccolo, ma si continua ad ottenere un centro quando la tartaruga si avvicina a meno di 10 passi: ancora come il vecchio bersaglio!

Per facilitare la variazione della grandezza del bersaglio, si può usare una variabile da inserire sia in TRACCIABERSAGLIO che in TIRO, oltre che in INIZIADATI, per fissarne il valore. La modifica, fatta così, è un poco più laboriosa, ma poi il bersaglio può essere ridotto o ingrandito molto più alla svelta. Supponiamo allora di chiamare la nuova variabile "RABE (abbreviazione di raggio bersaglio). Intanto la TRACCIABERSAGLIO deve avere anche questa fra i suoi dati di input:

```
PER TRACCIABERSAGLIO :XBERS :YBERS :RABE
SULAPENNA
VAXY :XBERS :YBERS
CERCHIETTO :RABE
FINE
```

Bisogna naturalmente modificare anche il richiamo di TRACCIABERSAGLIO in INCOMINCIA:

```
TRACCIABERSAGLIO :XBERS :YBERS :RABE
```

Va anche modificata la quinta istruzione della TIRO, che controlla se si è fatto centro:

```
VERIFICA DISTANZA :XBERS :YBERS< :RABE
```

In INIZIADATI bisogna poi aggiungere l'assegnazione del valore iniziale al raggio del bersaglio:

```
ASSEGNA "RABE 5
```

A questo punto, basta cambiare il valore assegnato in INIZIADATI, e il gioco è fatto!

10.6 Per un gioco più facile

Alcuni, almeno all'inizio, possono trovare difficile già la prima versione di TIRO.A.SEGNO e giocherebbero più volentieri con una versione facilitata. Il modo più semplice è forse l'impiego di un bersaglio più grande: basta assegnare valori più grandi al raggio del bersaglio, ad esempio 15 o 20; se poi si è già realizzata la nuova versione secondo le indicazioni del paragrafo 10.5, l'unica cosa da fare è assegnare il nuovo valore nell'istruzione di INIZIADATI che riguarda "RABE.
Meglio ancora sarebbe poi realizzare un gioco che si comporti in modo intelligente, dando qualche suggerimento utile al tiratore. Per fare questo il programma deve però memorizzare qualche informazione in più, ad esempio la distanza e l'angolo fra la tartaruga e il bersaglio, da paragonare con la distanza e l'angolo scelti dal giocatore per il suo colpo.
Ecco come si può fare. Si calcola la distanza e l'angolo fra il bersaglio e la tartaruga con la procedura di servizio DISTANZA e il comando VERSO del Logo. Si aggiungono inoltre in INCOMINCIA due nuove variabili, "DISBE e "ANGOBE, per memorizzare questi due nuovi valori.
Ecco le istruzioni da aggiungere alla INCOMINCIA:

```
ASSEGNA "DISBE DISTANZA :XBERS :YBERS
ASSEGNA "ANGOBE VERSO :XBERS :YBERS
```

La prima istruzione assegna il nome "DISBE al valore della distanza del bersaglio dalla posizione di partenza della tartaruga. La seconda richiama il comando VERSO, il cui output è il valore dell'angolo fra il bersaglio e la tartaruga, a cui viene assegnato il nome "ANGOBE. Durante il gioco, il computer paragonerà questi due valori con la distanza e l'angolo scelti dal giocatore per il suo colpo. Ecco come diventa INCOMINCIA:

```
PER INCOMINCIA
NASTARTA
TRACCIABERSAGLIO :XBERS :YBERS :RABE
TARTARUGAVAI :XPART :YPART :OPART
ASSEGNA "DISBE DISTANZA :XBERS :YBERS
ASSEGNA "ANGOBE VERSO :XBERS :YBERS
MOSTARTA
FINE
```

Ci vuole ancora un'altra variabile per memorizzare l'angolo scelto dal giocatore per il suo colpo (la possiamo chiamare "ANGOLPO): va inserita in TIRO (ricordiamo che qui c'è già la variabile "GITTATA, che memorizza la distanza alla quale è tirato il colpo). Ecco le modifiche:

```
PER TIRO
ASSEGNA "COLPI :COLPI+1
ASSEGNA "ANGOLPO DIREZIONE
STAMPA [A CHE DISTANZA?] ASSEGNA "GITTATA LEGGINUMERO
GIULAPENNA AVANTI :GITTATA
VERIFICA DISTANZA :XBERS :YBERS< :RABE
SEVERO COLPITO
SEFALSO MANCATO
FINE
```

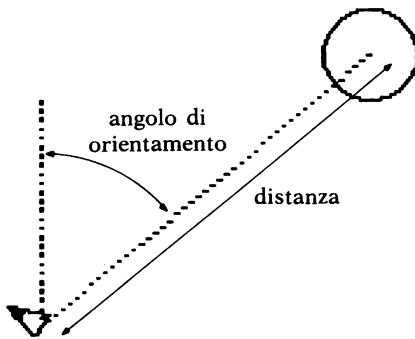



Figura 10.6 Due nuove variabili, "DISBE e "ANGOBE, memorizzano la distanza e l'angolo fra bersaglio e tartaruga (nella posizione di partenza)

Rimane il tocco finale: si aggiunge a MANCATO il richiamo di una nuova procedura, chiamata GUIDA, che chiede al giocatore se vuole aiuto: in tal caso, lo guida nella decisione del prossimo colpo.
Ecco la modifica:

```
PER MANCATO
STAMPA FRASE [MANCATO! COLPI TIRATI:] :COLPI
ASPETTA 200
TARTARUGAVAI :XPART :YPART :OPART
GUIDA
FINE
```

Ecco GUIDA:

```
PER GUIDA
STAMPA [VUOI UN CONSIGLIO PER IL PROSSIMO COLPO?]
STAMPA [PER FAVORE, RISPONDI SI O NO]
ASSEGNA "GUIDAMI LEGGIRIG
SE :GUIDAMI = [SI] TIGUIDO STOP
```

```
SE :GUIDAMI = [NO] STOP
GUIDA
FINE
```

Con l'ultima istruzione (dove c'è il richiamo a una nuova GUIDA) si fa in modo che, se il giocatore risponde qualcosa di diverso da SI o NO, il computer presenti ancora la domanda.
Ora vediamo la parte più interessante, cioè la nuova procedura TIGUIDO. È questa la "mente" vera del gioco, che paragona il colpo tirato con quello che avrebbe dovuto essere. Si riesce a capire come funziona?

```
PER TIGUIDO
VERIFICA :GITTATA>(:DISBE + :RABE)
SEVERO STAMPA [ACCORCIA UN PO']
VERIFICA :GITTATA<(:DISBE-:RABE)
SEVERO STAMPA [ALLUNGA UN PO']
VERIFICA :ANGOLPO>(:ANGOBE+5)
SEVERO STAMPA [MIRA PIÙ A SINISTRA]
SEVERO STAMPA [CIOE'...MENO A DESTRA]
VERIFICA :ANGOLPO<(:ANGOBE-5)
SEVERO STAMPA [MIRA PIÙ A DESTRA]
SEVERO STAMPA [CIOE'...MENO A SINISTRA]
FINE
```

Come si vede, la TIGUIDO considera un intervallo di possibili distanze e uno di possibili angoli, e non scrive nulla se il valore del colpo cade all'interno di tale intervallo. Nel caso visto, l'intervallo di distanza vale il raggio del bersaglio (in più e in meno) e l'intervallo di angolo vale 5 gradi (in più e in meno).
Funziona questa guida? Provate con un giocatore alle prime armi. Tenete conto che l'esempio proposto non va inteso in senso assoluto: anche se la guida non è un gran che, è pur sempre una buona occasione per pensare alle cose che deve "sapere" e fare una buona guida.

10.7 Come aggiungere una presentazione e modificare i messaggi

Alcuni apprezzano molto un gioco che si presenta da solo, descrivendo le regole e il modo di giocare. Se si vuole, si può aggiungere a TIRO.A.SEGNO una nuova procedura (si può chiamare MIPRESENTO) che presenta il gioco. Non è sempre facile preparare una buona presentazione, perché spesso non si capisce fino a che punto vale la pena di spingersi. E poi,

molto dipende dal gusto personale. Bisogna comunque fare attenzione a non superare i quaranta caratteri per riga (compresi gli spazi vuoti) e conviene fare in modo che le scritte si possano leggere con facilità, magari alternandole con righe vuote (basta ricorrere a STAMPA[ ], e si stampa una lista vuota).
In genere, è bene lasciare a chi gioca la scelta di far comparire o no la presentazione: si può usare la procedura ASPETTO, già vista nel Capitolo 9, alla fine di ciascuna pagina, per dare il tempo al giocatore di pensare:

```
PER ASPETTO
STAMPA [PER FAVORE, BATTI IL TASTO RETURN]
STAMPA LEGGIRIG
FINE
```

Abbiamo già visto che, in realtà, è il comando LEGGIRIG che aspetta finché non viene battuto almeno un tasto della tastiera (e/o il tasto RETURN).
Per modificare i messaggi scritti dal computer durante il gioco, bisogna riprendere le procedure che li scrivono: TIRO, COLPITO, MANCATO e, se ci sono, GUIDA e TIGUIDO. Si può scrivere quello che si vuole, naturalmente, guidati dal proprio gusto.
Scritte come:

```
SCEMO! HAI SBAGLIATO
```

piacciono ad alcuni; ad altri, per fortuna, no. In genere, le scritte che sembrano più originali diventano, col tempo, stucchevoli.

10.8 Come riunire le versioni del gioco

È facile, quando si fanno modifiche a un programma, disporre dopo un po' di versioni diverse. Anche TIRO.A.SEGNO seguirà probabilmente la stessa sorte, e si avrà così la versione originale, quella con gli effetti speciali, quella più facile, quella più impegnativa, e così via. Proviamo ora a riunirle in un "megagioco" che fa scegliere al giocatore la versione da usare. Si aggiunge allora una procedura all'inizio, che possiamo chiamare SCELTA, richiamandola per prima in VIA:

```
PER VIA
SCELTA
DISEGNA
INIZIADATI
INCOMINCIA
```

```
FINE
PER SCELTA
ESCIDIS
STAMPA [BENVENUTO AL TIRO A SEGNO]
STAMPA [VUOI SAPERE COME SI GIOCA?]
ASSEGNA "PRESENTATI LEGGIRIG
SE :PRESENTATI = [SI] MIPRESENTO
CHELIVELLO
FINE
```

Ecco come si può presentare la scelta del livello di difficoltà, con CHELIVELLO:

```
PER CHELIVELLO
PULISCITESTO
STAMPA [A CHE LIVELLO VUOI GIOCARE?]
STAMPA [ ]
STAMPA [1-FACILE: IL COMPUTER TI AIUTA]
STAMPA [2-COSI' COSI']
STAMPA [3-DIFFICILE]
STAMPA [ ]
STAMPA [PUOI SCEGLIERE 1, 2 OPPURE 3]
STAMPA [POI BATTI IL TASTO RETURN, PER FAVORE]
ASSEGNA "ILLIVELLO LEGGINUMERO
SE :ILLIVELLO = 1 ASSEGNA "GIOCATORE "SCHIAPPA STOP
SE :ILLIVELLO = 2 ASSEGNA "GIOCATORE "NORMALE STOP
SE :ILLIVELLO = 3 ASSEGNA "GIOCATORE "DRAGO STOP
CHELIVELLO
FINE
```

TRABOCCHETTO

Attenti a non confondere i nomi delle variabili con i loro valori. A volte si cade in errore, se per esempio il valore è a sua volta una parola. Qui "SCHIAPPA, "NORMALE e "DRAGO non sono nomi di variabile, ma valori assegnati alla variabile chiamata "GIOCATORE.

Con la CHELIVELLO, il computer stabilisce il valore della variabile "GIOCATORE, che può essere usata per scegliere la versione adatta, con il giusto livello di difficoltà. Si modifica la INIZIADATI aggiungendo tre istruzioni:

```
SE :GIOCATORE = "SCHIAPPA ASSEGNA "RABE 20
SE :GIOCATORE = "NORMALE ASSEGNA "RABE 10
SE :GIOCATORE = "DRAGO ASSEGNA "RABE 5
```

Se è stato scelto il livello più facile, conviene che il computer offra un aiuto, quando il colpo è mancato. Si aggiunge allora un'istruzione a MANCATO:

```
PER MANCATO
STAMPA [MANCATO! COLPI TIRATI:] :COLPI
ASPETTA 200
RIPARTI :XPART :YPART :OPART
SE :GIOCATORE = "SCHIAPPA GUIDA
FINE
```

Viene cioè richiamata GUIDA, già vista nel paragrafo 10.6, che richiama poi TIGUIDO.
La INCOMINCIA deve inoltre comprendere le due istruzioni già viste nel paragrafo 10.6, che assegnano i valori alle variabili "DISBE e "ANGOBE. Se il livello scelto è quello intermedio, non conviene aggiungere altro. Se invece viene scelto il livello più difficile, si può far cancellare la traiettoria appena tracciata (come si è visto nel paragrafo 10.5, con ASCOL −1 nella procedura RIPARTI):

```
PER TIRO
ASSEGNA "COLPI :COLPI+1
ASSEGNA "ANGOLPO DIREZIONE
STAMPA [A CHE DISTANZA?]
ASSEGNA "GITTATA LEGGINUMERO
GIULAPENNA AVANTI :GITTATA
VERIFICA DISTANZA :XBERS :YBERS<:RABE
SEVERO COLPITO BANG 20
SEFALSO MANCATO
FINE
```

e ancora:

```
PER RIPARTI :XPART :YPART :OPART
SE :GIOCATORE = "DRAGO ALLORA ASCOL -1 ALTRIMENTI
```

```
SULAPENNA
VAXY :XPART :YPART
ASDIR :OPART
ASCOL 1
FINE
```

TRABOCCHETTO

Le modifiche sono un affar serio, se si ha a che fare con un programma complesso come TIRO.A.SEGNO, e non vanno mai prese sottogamba: si può star certi che ci si dimentica sempre di qualcosa. L'unica cosa da fare quando si modifica un programma, è provarlo a lungo per vedere se funziona a dovere.
Quando si prova un gioco, bisogna cercare di provare tutte le possibili situazioni, andando ogni volta fino in fondo. L'ideale è un amico che vuole giocare: fate fare a lui le prove (ma attenzione, senz'altro avrà le sue idee e proporrà nuove modifiche!).

11

SCHIZZO

Abbiamo visto nel Capitolo 3 come si usa SCHIZZO. Esso è stato realizzato principalmente per i piccoli, che così possono disegnare con la tartaruga e incominciare l'esplorazione del Logo senza dover battere i comandi completi del Logo e i loro parametri numerici.
Va da sé che programmi di questo tipo sono utili anche a chi non riesce a battere i comandi alla tastiera, per un qualche difetto fisico.
SCHIZZO è uno strumento interattivo, così come sono interattivi i quiz del Capitolo 9 e il gioco TIRO.A.SEGNO del Capitolo 10. In più, esso ha comandi molto semplici, che richiedono la battuta di un solo tasto (appena si batte il tasto, subito si innesca l'azione che ad esso corrisponde, senza dover battere anche il tasto RETURN): per un principiante, è un bel vantaggio.
Ah! Ma allora è possibile cambiare il modo in cui si usa il Logo, con i comandi stessi del Logo! È vero, si può fare.
Molti si domandano a che età si può imparare il Logo: in realtà, non c'è un limite preciso, perché il Logo è adattabile all'età di chi lo impara. Basta solo decidere come lo si vorrebbe, e scrivere una procedura in Logo che traduca in pratica la nostra idea. In breve, il Logo consente di costruire un ambiente educativo su misura di chi impara.
Volendo, si può dare SCHIZZO a un bambino, osservare come lo usa e adattarlo ai suoi bisogni e ai suoi gusti, costruendo pian piano un ambiente su misura per l'apprendimento del Logo. Più avanti, in questo capitolo, è presentato il modo per modificare il programma.

Comando	Abbreviazione	Esempi
PUNTOACAPO		SE :COMAN = "Z PUNTOACAPO
COSA		STAMPA COSA :PRED

Si usano i file "SCHIZZO, "LEGGITASTO e "LEGGINUMERO dall'Appendice A

Nuove procedure di servizio usate:

Nome	Esempio
LEGGITASTO	ASSEGNA "COMAN LEGGITASTO

11.1 Come funzionano le procedure di SCHIZZO

Prima di proseguire, conviene rileggere il paragrafo 3.2 "SCHIZZO, ovvero come disegnare comodamente", per rinfrescare la memoria sul suo impiego (magari, con qualche prova pratica). Si deve poi caricare il file "SCHIZZO e il file "LEGGITASTO (se non lo si è già fatto, si ricopiano dall'Appendice A).
Incominciamo con un programma di disegno molto, molto semplificato, con l'idea di completarlo man mano fino ad ottenere la SCHIZZO vista nel Capitolo 3. Poi andremo oltre, con estensioni e miglioramenti vari.
Un programma di disegno deve reagire alla battuta dei tasti previsti: se si batte A, la tartaruga deve spostarsi un po' in avanti, con D ruotare un po' a destra e con S ruotare un po' a sinistra. Volendo finire, si batte Z (per la scelta dei tasti, che è arbitraria, ci si può basare, come in questo caso, sul loro significato mnemonico; altri preferiscono raggruppare i comandi in tasti vicini, sulla tastiera. Con la scelta appena fatta, ci si avvicina di molto ai comandi normali del Logo, e questo può servire a un piccolo principiante). Ecco da dove partiamo:

```
PER SCHIZZO
COMANDO
SCHIZZO
FINE
PER COMANDO
ASSEGNA "COMAN LEGGITASTO
SE :COMAN = "A AVANTI 20
SE :COMAN = "D DESTRA 30
SE :COMAN = "S SINISTRA 30
SE :COMAN = "Z PUNTOACAPO
FINE
```

Si può battere DISEGNA (per ripulire lo schermo) e poi usare questa SCHIZZO, che è molto semplice. Essa richiama COMANDO per avere il nuovo comando di chi sta usando il computer, e poi chiama una nuova procedura SCHIZZO. Anche la COMANDO è semplice: la prima istruzione assegna il nome "COMAN a ciò che viene battuto alla tastiera (un tasto solo). Le istruzioni successive verificano se il tasto battuto è uno di quelli previsti (A, D, S o Z), per innescare l'azione conseguente (se no, non succede niente). Se :COMAN è una delle lettere previste, viene eseguito il resto dell'istruzione (che sposta la tartaruga in avanti, o la fa ruotare, o arresta il programma). Qui c'è un comando nuovo del Logo, PUNTOACAPO; esso arresta tutte le procedure (non solo la procedura in cui compare, come fa invece il comando STOP). Dopo PUNTOACAPO, compare il simbolo ? del Logo e si ritorna a dover battere i comandi normali.

TRABOCCHETTO

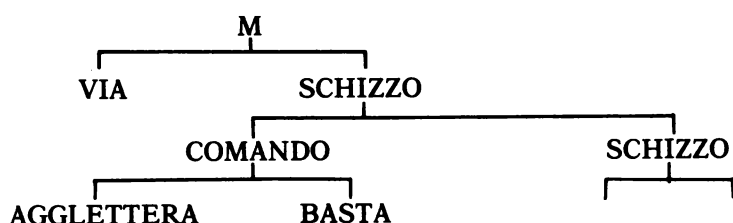
Il comando PUNTOACAPO non funziona come STOP, ma fa di più. Con STOP si arresta infatti solo la procedura che al momento è attiva, e non quelle che sono in attesa. Se al posto di PUNTOACAPO ci fosse STOP, nell'istruzione:

```
SE :COMAN = "Z STOP
```

la SCHIZZO non si fermerebbe. Verrebbe arrestata solo COMANDO, mentre SCHIZZO passerebbe alla sua istruzione successiva, con il richiamo di una nuova SCHIZZO, che chiama una nuova COMANDO, e così via. Vale la pena di provare! Questo è un bug di quelli difficili da scovare!

11.2 E se SCHIZZO ricordasse cosa ha fatto?

Con la SCHIZZO appena vista, le figure vengono tracciate e poi perse, quando lo schermo è ripulito. Sarebbe bello poter dare un nome ai disegni fatti, per poterli rivedere in seguito, riproducendo ciò che si è già fat-

to (magari, con un comando solo). La cosa si può fare, e in molti modi. Ad esempio, la SCHIZZO può memorizzare i comandi dati (che poi sono semplici lettere, fra quelle previste) in una lista, alla quale viene assegnato un nome. Richiamando poi (con una nuova procedura) la lista per nome, si ottiene di nuovo il disegno.
Per far funzionare SCHIZZO come nel Capitolo 3, bisogna completarla con una nuova superprocedura (che si può chiamare come si vuole, anche semplicemente M) e con alcune altre procedure particolari, come si vede nel seguente schema di connessione:

La M e le procedure da essa richiamate (SCHIZZO e VIA) sono abbastanza semplici:

```
PER M
VIA
SCHIZZO
FINE
```

La VIA prepara il computer per la SCHIZZO (attenzione, per usare d'ora in poi SCHIZZO bisogna battere M):

```
PER VIA
ASSEGNA "LISCOM [ ]
DISEGNA
FINE
```

La lista "LISCOM è destinata a contenere i comandi battuti man mano, e parte vuota, come si vede. La VIA ripulisce anche lo schermo. La SCHIZZO rimane così com'è:

```
PER SCHIZZO
COMANDO
SCHIZZO
FINE
```

Cambia invece la COMANDO, che d'ora in poi richiama altre due proce-

dure, AGGLETTERA (che aggiunge i comandi alla lista "LISCOM) e BASTA (che termina lo schizzo, assegnando un nome scelto da chi lo ha fatto). Viene anche aggiunta la possibilità di un nuovo movimento all'indietro (con I):

```
PER COMANDO
ASSEGNA "COMAN LEGGITASTO
SE :COMAN = "A AVANTI 20 AGGLETTERA :COMAN
SE :COMAN = "I INDIETRO 20 AGGLETTERA :COMAN
SE :COMAN = "D DESTRA 30 AGGLETTERA :COMAN
SE :COMAN = "S SINISTRA 30 AGGLETTERA :COMAN
SE :COMAN = "Z BASTA PUNTOACAPO
FINE
PER AGGLETTERA :LET
ASSEGNA "LISCOM FRASE :LISCOM :LET
FINE
```

La AGGLETTERA aggiunge il nuovo comando (ricorrendo a FRASE) alla lista :LISCOM.
Ecco BASTA:

```
PER BASTA
DISCHERMO
STAMPA [COME CHIAMI IL DISEGNO?]
STAMPA [(UNA SOLA PAROLA)]
STAMPA [SE LO VUOI BUTTARE,]
STAMPA [BATTI SOLO IL TASTO RETURN]
ASSEGNA "RISPOS LEGGIRIG
SE :RISPOS = [ ] STOP
ASSEGNA (PRIMO :RISPOS) :LISCOM
FINE
```

Ecco qui l'ultima istruzione di BASTA, che può sembrare un pochino strana:

```
ASSEGNA (PRIMO :RISPOS) :LISCOM
```

Finora abbiamo visto usare il comando ASSEGNA in modo diverso, con un nome come primo dato e un valore per secondo dato, come in questo esempio:

```
ASSEGNA "ANGOLPO DIREZIONE
```

Qui ASSEGNA riceve il suo primo dato di input da :RISPOS, che è la li-

sta (composta da una parola sola) appena battuta alla tastiera (il comando LEGGIRIG ha sempre in output una lista). Dato che ASSEGNA vuole in input una parola, come primo dato, si ricorre al comando PRIMO, che estrae il primo elemento della lista :RISPOS (una parola, appunto) e lo passa a ASSEGNA. Il secondo dato di ASSEGNA è la lista :LISCOM, dei comandi dati per tracciare il disegno.

SUGGERIMENTO

C'è da capire la differenza fra una parola e una lista, magari formata da una sola parola. È per questa differenza che va usato il comando PRIMO in BASTA. Provate questi casi, e vi renderete conto da soli della differenza:

```
STAMPA [BUONGIORNO]
BUONGIORNO
STAMPA "BUONGIORNO
BUONGIORNO
STAMPA "BUONGIORNO = [BUONGIORNO]
FALSO
```

Proprio così, anche se i due [BUONGIORNO] e "BUONGIORNO sembrano la stessa cosa, non lo sono per il computer. Lo diventano con il comando PRIMO:

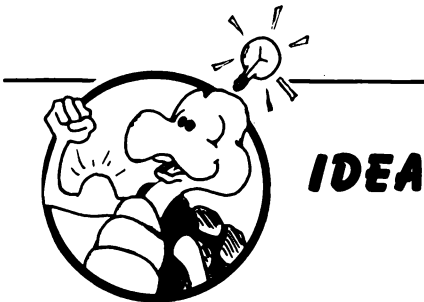
```
STAMPA PRIMO [BUONGIORNO]
BUONGIORNO
STAMPA "BUONGIORNO
BUONGIORNO
STAMPA "BUONGIORNO = PRIMO [BUONGIORNO]
VERO
```

Eh sì, "BUONGIORNO è lo stesso che PRIMO [BUONGIORNO] (il primo elemento di una lista è una parola: ecco perché).
Ora dovrebbe essere più chiaro il comando:

```
ASSEGNA (PRIMO :RISPOS) :LISCOM
```

RIPRENDIAMO UN DISEGNO

Per riprendere un disegno, memorizzato nel modo appena visto, c'è la procedura RD, che ha come dato di input la lista :LISCOM.
Ecco qua:

```
PER RD :LISCOM
SE :LISCOM = [ ] STOP
RICOMANDO PRIMO :LISCOM
RD MENPRI :LISCOM
FINE
```

La RD è una classica procedura di elaborazione di liste. Ha uno schema ripetitivo, molto popolare nella programmazione in Logo (lo si vedrà ancora applicato nelle procedure di servizio descritte nel Capitolo 14). Lo schema di base è descritto qui di seguito.
La RD ha in input una lista. Per prima cosa controlla se è vuota (e se lo è si ferma). Se la lista non è vuota, chiama la RICOMANDO alla quale assegna il primo elemento della lista stessa.
Quando la RICOMANDO termina il suo compito, viene richiamata una nuova RD, ma questa si trova in input la lista di prima meno il suo primo elemento (è l'effetto del comando MENPRI). Come si vede, man mano che sono richiamate nuove RD, la lista LISCOM si riduce sempre di più, fino a svuotarsi completamente. Ma allora l'ultima RD si ritrova in input una lista vuota, e si ferma.

La RICOMANDO, che è chiamata da RD, esegue il comando che questa le passa. Essa è come COMANDO, ma non riceve l'input da tastiera, bensì dalla lista :LISCOM (per mezzo del comando PRIMO, che ne estrae il primo elemento):

```
PER RICOMANDO :COMAN
SE :COMAN = "A AVANTI 20
SE :COMAN = "I INDIETRO 20
SE :COMAN = "D DESTRA 30
SE :COMAN = "S SINISTRA 30
FINE
```

Se si è finito un disegno e lo si memorizza, come si è visto prima, ad esempio con il nome "CASA, per riprenderlo si batte:

```
RD :CASA
```

come si è visto nel Capitolo 3. Attenzione all'input di RD, che deve essere una lista. Ecco perché si deve indicare :CASA, e non "CASA, che è il nome della lista (cioè, una parola).

SUGGERIMENTO

SCHIZZO offre una nuova opportunità per capire bene la differenza fra un oggetto e il suo nome. Un modo pratico di far capire la differenza fra la lista dei comandi e il nome a questa associato è seguire la sua costruzione passo passo. Si può far stampare al computer il valore di :LISCOM ad ogni nuovo inserimento, e alla fine, quando il disegno è completo. Si deve aggiungere questa istruzione a SCHIZZO, subito dopo COMANDO:

```
STAMPA FRASE [:LISCOM VALE ORA] :LISCOM
```

Va aggiunta inoltre alla fine di BASTA:

```
STAMPA (FRASE PAROLA ":PRIMO :RISPOS [VALE] :LISCOM)
```

Queste istruzioni non modificano in alcun modo il funzionamento delle procedure: il loro scopo è solo didattico e possono essere tolte quando non servono più.

11.3 Miglioriamo SCHIZZO

Qui si presentano alcune possibili modifiche a SCHIZZO. Certamente il lettore ne penserà molte altre da solo, specie se ha un piccolo che usa il programma e che anzi va incoraggiato a dire la sua. Ecco qui alcuni suggerimenti:

1. Aggiungere nuovi comandi, oltre a COMANDO e RICOMANDO. Si può ad esempio inserire il programma CERCHI (caricandolo da disco), e aggiungere questa istruzione a COMANDO:

   ```
   SE :COMAN = "C CERCHIODES 10 AGGLETTERA :COMAN
   ```

   Anche in RICOMAN va aggiunta un'istruzione:

   ```
   SE :COMAN = "C CERCHIODES 10
   ```

   Se si ha un video a colori, ci si può sbizzarrire coi colori, che possono essere indicati battendo semplicemente un numero. Ecco le istruzioni per COMANDO:

   ```
   SE :COMAN = 0 ASCOL 0 AGGLETTERA :COMAN
   SE :COMAN = 1 ASCOL 1 AGGLETTERA :COMAN
   SE :COMAN = 2 ASCOL 2 AGGLETTERA :COMAN
   ```

   Aggiunte corrispondenti vanno fatte in RICOMANDO.
   Si possono anche considerare abbreviazioni per SULAPENNA e GIULAPENNA:

   ```
   SE :COMAN = "U SULAPENNA AGGLETTERA :COMAN
   SE :COMAN = "G GIULAPENNA AGGLETTERA :COMAN
   ```

2. Realizzare un comando per ripulire lo schermo e per vuotare la lista :LISCOM, se il disegno non piace, per ricominciare con un altro. Ecco l'istruzione per COMANDO:

   ```
   SE :COMAN = "V VIA
   ```

   La procedura VIA, già chiamata in M, riparte da zero. Da notare che qui non si deve più chiamare AGGLETTERA, perché anzi la lista :LISCOM va svuotata. Non va toccata neanche la RICOMANDO. Battendo V, si riparte da capo.

3. Aggiungere un disegno già pronto (che cioè ha già un nome suo) al disegno appena tracciato. Ci vuole una nuova istruzione in COMANDO:

```
SE :COMAN = "P PRENDISEGNO
```

e una nuova procedura:

```
PER PRENDISEGNO
STAMPA [CHE DISEGNO VUOI PRENDERE?]
ASSEGNA "PRED PRIMO LEGGIRIG
RD COSA :PRED
AGGLETTERA COSA :PRED
FINE
```

La nuova PRENDISEGNO va spiegata: COSA è un comando Logo che fornisce in output il valore associato al nome fornito in input. Quando si batte il nome del disegno da prendere, la seconda istruzione di PRENDISEGNO assegna a questo un nuovo nome, cioè "PRED:

```
ASSEGNA "PRED PRIMO LEGGIRIG
```

RD e AGGLETTERA vogliono in input un lista, e :PRED è una parola che contiene il nome di una lista. Allora, COSA :PRED è la lista che ha il nome segnato in :PRED.
Per capire meglio, conviene provare in modo diretto. Si prepara una lista di comandi:

```
ASSEGNA "PROVA [A A D D A]
```

Poi si assegna al suo nome il nome "PRED

```
ASSEGNA "PRED "PROVA
```

Questo è ciò che accade nella seconda istruzione di PRENDISEGNO. Ora si può battere:

```
STAMPA :PRED
PROVA
STAMPA COSA "PROVA
A A D D A
STAMPA :PROVA
A A D D A
STAMPA COSA :PRED
A A D D A
```

Le ultime tre istruzioni stampano le stesse lettere, perché :PRED vale proprio "PROVA, e quindi COSA "PROVA, COSA :PRED e :PROVA sono modi diversi di chiamare la lista [A A D D A].

Non importa se ora non si capisce del tutto questo punto (che non è facile, dato che si tratta di un nome che ne nomina un altro): bisogna però controllare che PRENDISEGNO funzioni a dovere.

4. Rendere variabili la distanza e l'angolo usati nelle procedure COMANDO e RICOMANDO. Bisogna modificare tutte le istruzioni in cui la tartaruga è spostata o ruotata, e aggiungere a VIA le istruzioni che fissano il valore dell'avanzamento e della rotazione:

```
PER VIA
ASSEGNA "DISTANZA 20
ASSEGNA "ANGOLO 30
ASSEGNA "LISCOM [ ]
DISEGNA
FINE
PER COMANDO
ASSEGNA "COMAN LEGGITASTO
SE :COMAN = "A AVANTI :DISTANZA AGGLETTERA :COMAN
SE :COMAN = "I INDIETRO :DISTANZA AGGLETTERA :COMAN
SE :COMAN = "D DESTRA :ANGOLO AGGLETTERA :COMAN
SE :COMAN = "S SINISTRA :ANGOLO AGGLETTERA :COMAN
SE :COMAN = "C CERCHIODES :DISTANZA AGGLETTRA :COMAN
SE :COMAN = "Z BASTA PUNTOACAPO
FINE
PER RICOMANDO :COMAN
SE :COMAN = "A AVANTI :DISTANZA
SE :COMAN = "I INDIETRO :DISTANZA
SE :COMAN = "D DESTRA :ANGOLO
SE :COMAN = "S SINISTRA :ANGOLO
SE :COMAN = "C CERCHIODES :DISTANZA
FINE
```

Fatto ciò si possono poi fissare come si vuole i valori assegnati nelle prime due istruzioni di VIA.
Volendo, si può anche fare in modo che la distanza e l'angolo vengano modificati senza neanche vedere la VIA. Bisogna preparare tre procedure nuove, che chiameremo CAMBIA, CHEDIS e CHEANG. Prima però modifichiamo VIA, con l'istruzione che richiama CAMBIA:

```
PER VIA
ASSEGNA "DISTANZA 20
ASSEGNA "ANGOLO 30
ASSEGNA "LISCOM [ ]
DISEGNA
CAMBIA
```

```
PULISCITESTO
FINE
```

Ecco CAMBIA:

```
PER CAMBIA
STAMPA (FRASE [LA TARTARUGA FA] :DISTANZA [PASSI])
STAMPA [SE LI VUOI CAMBIARE, BATTI S]
SE LEGGIRIG = [S] CHEDIS
STAMPA [ ]
STAMPA (FRASE [E RUOTA DI] :ANGOLO [GRADI])
STAMPA [SE LO VUOI CAMBIARE, BATTI S]
SE LEGGIRIG = [S] CHEANG
FINE
```

Come si vede, vengono richiamate CHEANG e CHEDIS, per fissare nuovi valori:

```
PER CHEDIS
STAMPA [QUANTI PASSI VUOI FARE?]
ASSEGNA "DISTANZA LEGGINUMERO
STAMPA (FRASE [LA TARTARUGA FA] :DISTANZA [PASSI])
FINE
PER CHEANG
STAMPA [DI QUANTI GRADI VUOI RUOTARE?]
ASSEGNA "ANGOLO LEGGINUMERO
STAMPA (FRASE [LA TARTARUGA RUOTA DI] :ANGOLO [GRADI])
FINE
```

Abbiamo già incontrato LEGGINUMERO: è una procedura di servizio, descritta nel Capitolo 14.
Così ogni volta che si incomincia a disegnare (battendo M), il computer ci fa scegliere se tenere o cambiare il passo e la rotazione della tartaruga. Se non li si vuole cambiare, si batte RETURN. Ancora una cosa: se si ha il comando abbreviato V (che abbandona il disegno) in COMANDO:

```
SE :COMAN = "V VIA
```

conviene modificarlo:

```
SE :COMAN = "V DISEGNA ASSEGNA "LISCOM [ ]
```

In tal modo si conservano la distanza e l'angolo, quando si cancella il disegno per ricominciare da capo.

12

La tartaruga in corsa

I giochi di animazione sono una delle attività più stimolanti da fare con il computer. Qui si considera il modo di realizzarne uno, abbastanza semplice, basato su una tartaruga animata.
Per essere animato, un oggetto (come la tartaruga) si deve poter muovere in modo che sembri vivo, come se fosse un animale, appunto. Per animare la tartaruga, ci vuole una procedura che la tenga in movimento fino a quando non viene battuto alla tastiera un comando che la ferma, o le fa cambiar direzione, o altro ancora. È facile scrivere una procedura che tiene in movimento la tartaruga: basta prendere uno dei molti esempi di disegno con la tartaruga visti nei capitoli precedenti, e fare qualche lieve modifica. Il controllo della tartaruga con la semplice battuta di un tasto, poi, è già stato descritto nel Capitolo 11, con il programma SCHIZZO. Insomma, per avere una tartaruga animata basta combinare cose già fatte.
Già di per sé la tartaruga animata è divertente, ma lo diventa ancor più se intorno si riesce a costruire una storia, o per lo meno una situazione. Qui si presenta la tartaruga che corre in pista, come un'auto da corsa: il gioco può naturalmente essere arricchito con procedure di contorno, che controllano ad esempio se la tartaruga esce di pista, e segnalano quando essa taglia il traguardo.

Comando	Abbreviazione	Esempi
TUTTIVERI?		SE TUTTIVERI? (YCOR>0) (XCOR>0) STOP
XOR		STAMPA XCOR, VAX XCOR+20
YCOR		STAMPA YCOR, VAY YCOR−20
RIPORTA	RI	RIPORTA "FALSO, RI :LUNGH+10

Vengono usati i file "LEGGITASTO,"CERCHIETTO,"DISTANZA,"RETTANGOLI e "CORSA dell'Appendice A.

Nuove procedure di servizio usate:

Nome	**Esempio**
TRETTAN	TRETTAN 0 0 50 30
DENTRO?	SE (DENTRO? 0 0 50 30) STOP
FUORI?	SE (FUORI? (-20) (-20) 70 50) STOP

12.1 La tartaruga animata

Ecco una procedura semplice per animare la tartaruga (si chiama PILOTA perché poi verrà usata nel gioco che serve da esempio, che riguarda una corsa in pista):

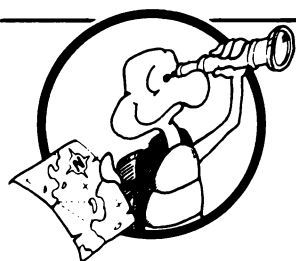
```
PER PILOTA :META
AVANTI :META
PILOTA :META
FINE
```

Se si batte PILOTA 1, la tartaruga si mette in moto, spingendosi in avanti un passo per volta. Con PILOTA 10, la si vede correre dieci volte più veloce (perché fa salti più lunghi).
Certo, se la tartaruga va solo avanti non diverte gran che: bisogna anche farle cambiare direzione. Si aggiunge allora un'altra procedura, che qui ha nome COMANDO, la quale accetta alcuni comandi da tastiera, come vedremo subito:

```
PER PILOTA :META
AVANTI :META
COMANDO
PILOTA :META
FINE
```

La COMANDO controlla se è stato battuto uno dei tasti previsti, per eseguire il comando corrispondente. Poi la nuova PILOTA sposta ancora in avanti la tartaruga. Se non si batte nessun tasto, la COMANDO si arresta senza aver fatto nulla, e la nuova PILOTA fa muovere la tartaruga nella stessa direzione di prima. In COMANDO si ricorre alla procedura di servizio LEGGITASTO, riportata in Appendice A:

```
PER COMANDO
ASSEGNA "COMAN LEGGITASTO
SE :COMAN = " STOP
SE :COMAN = "D DESTRA 30 STOP
SE :COMAN = "S SINISTRA 30 STOP
FINE
```

La prima istruzione assegna il nome "COMAN alla lettera battuta dal giocatore. La seconda istruzione non è un errore: il segno " da solo significa una parola vuota, cioè una parola senza lettere. Se non si batte nulla alla tastiera, LEGGITASTO ha in output proprio una parola vuota: in questo caso, COMANDO si arresta subito, senza fare più nulla. Il comando STOP alla fine delle altre istruzioni fa arrestare COMANDO subito dopo il riconoscimento di una delle lettere previste e l'esecuzione delle azioni conseguenti. Senza questi STOP, il computer dovrebbe fare più lavoro di quello necessario, impiegando anche più tempo. Con solo un paio di comandi, la cosa non ha importanza, ma quando se ne avranno di più sarà diverso. A questo punto si può pilotare la tartaruga dove si vuole: se si imposta un numero grande in PILOTA, la tartaruga corre veloce; con un numero basso, invece, va lenta e si lascia guidare più facilmente.

ESPLORAZIONE

Provate a disegnare con PILOTA. Riuscite a tracciare un cerchio? E che ne dite di un nome? Provate anche a disegnare qualche figura sullo schermo, e a muovere poi la tartaruga solo nella parte di schermo rimasta libera. La Figura 12.1 illustra qualche esempio.

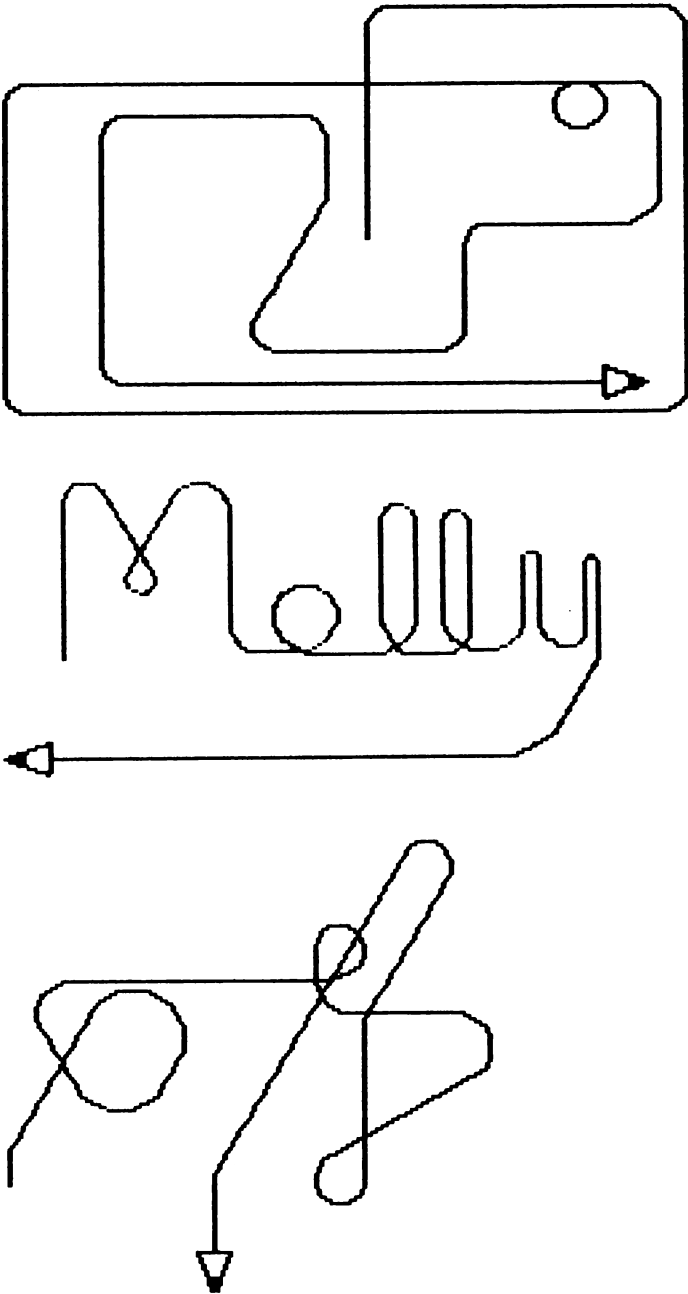



Figura 12.1 Disegni fatti con PILOTA

SUGGERIMENTO

Nella procedura COMANDO ci sono alcune finezze di programmazione. Per prima cosa, compare una parola vuota: il concetto di un oggetto vuoto è molto familiare ai matematici, ma per molta gente è invece astruso. In molti casi però (come nel nostro esempio) una parola (o una lista) vuota risulta molto utile: il caso tipico riguarda i comandi condizionali, quando si tratta di fermare una procedura in mancanza di certi eventi (nell'esempio, la COMANDO si arresta se non si batte nulla alla tastiera).
È possibile scrivere COMANDO senza preoccuparsi di ciò che viene o non viene battuto alla tastiera, e anche senza il comando STOP alla fine di ogni istruzione:

```
PER COMANDO
ASSEGNA "COMAN LEGGITASTO
SE :COMAN = "D DESTRA 30
SE :COMAN = "S SINISTRA 30
FINE
```

Per ora, in effetti, COMANDO è talmente ridotta che non si ottiene un gran vantaggio ad aggiungere un'istruzione e a completare con STOP le due esistenti. Quando però verrà il momento di ingrandire il progetto, la COMANDO avrà sempre più controlli da fare, e allora impiegherebbe troppo tempo, senza gli accorgimenti appena visti.
Qui si presenta un dilemma tipico: conviene incominciare con un esempio molto grezzo e aspettare le prime difficoltà pratiche per presentare certi accorgimenti, magari a costo di complicare la spiegazione? Se si insegna ad una persona, forse è meglio seguire il primo approccio, che consente fra l'altro di toccare con mano i motivi che giustificano gli accorgimenti (che dire dell'esperienza di vedere la tartaruga muoversi sempre più lentamente, mano a mano che si aggiungono nuovi controlli per nuovi possibili comandi?) In questo modo però è necessario che gli accorgimenti vengano presentati in modo tempestivo, non appena se ne avverte la convenienza.
In un libro è diverso: non si conoscono le reazioni del lettore, ed è forse meglio anticipare certi fattori critici, come la velocità di esecuzione dei comandi in un gioco di animazione. Se il lettore presenta a sua volta l'argomento ad altri, può darsi che gli convenga partire invece con l'esempio più grezzo.

12.2 Miglioriamo l'animazione

Ora che sappiamo pilotare la tartaruga in lungo e in largo, possiamo ampliare COMANDO, in modo da aumentare il controllo dei movimenti della tartaruga. Aggiungere nuovi comandi è facile: si decide quale lettera va battuta per attivarli e si aggiunge una nuova istruzione in COMANDO:

```
SE :COMAN = "G GIULAPENNA STOP
SE :COMAN = "U SULAPENNA STOP
SE :COMAN = "V ASSEGNA "META :META+1 STOP
SE :COMAN = "L ASSEGNA "META :META -1 STOP
```

Si può anche riservare qualche comando al controllo del colore, magari ricorrendo ai numeri:

```
SE :COMAN = 0 ASCOL 0 STOP
SE :COMAN = 1 ASCOL 1 STOP
ecc.
```

Che ne dite di un comando per far saltare la tartaruga? Eccolo qua:

```
SE :COMAN = "Z ZAP STOP
PER ZAP
SULAPENNA
DESTRA 90
AVANTI 100
SINISTRA 90
GIULAPENNA
FINE
```

Battendo Z, si ottiene il salto: oplà! Poi la tartaruga prosegue nella sua direzione. Un altro bell'effetto è la fermata improvvisa (senza arrestare la procedura):

```
SE :COMAN = "H ALTOLA STOP
PER ALTOLA
ASSEGNA "META 0
FINE
```

Per riprendere la corsa, basta battere V oppure L.
La scelta dei tasti merita attenzione: alcuni li scelgono per il loro significato mnemonico, altri invece preferiscono usare tasti disposti vicini sulla tastiera; in questo caso conviene, almeno all'inizio, disegnare un piccolo schema con l'indicazione dei tasti e del loro significato.

12.3 In corsa — Parte I

Per giocare alla corsa in pista con la tartaruga, il modo più facile è disegnare la pista e poi guidarvi la tartaruga con PILOTA. Si traccia allora la pista di Figura 12.2 (con la procedura CERCHI già descritta), si colloca la tartaruga al suo interno e poi si batte PILOTA.

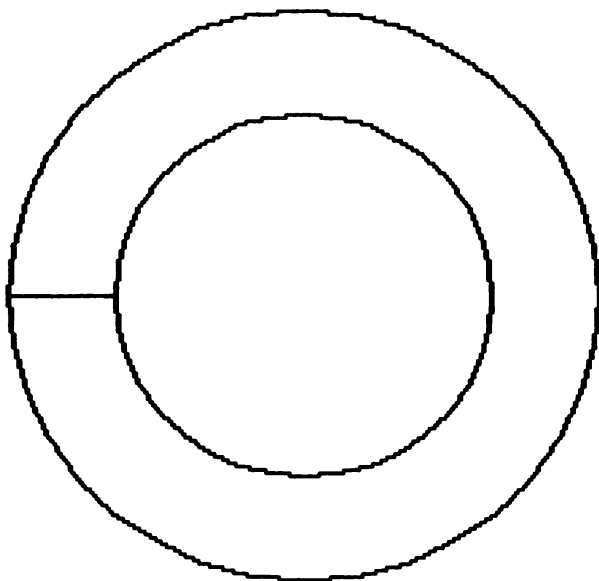



Figura 12.2 La pista ad anello

Così com'è, non è una gran corsa, perché il computer non controlla dove va la tartaruga: qualunque percorso va bene, anche se esce di pista. Il gioco migliora se si fa fermare la tartaruga quando esce di pista, oppure magari quando riesce a compiere un giro completo senza incidenti.
Comunque per ora, dato che il computer non controlla il percorso, qualunque pista va bene (con o senza ostacoli da evitare, angoli, chicane, cul de sac, ecc.).
Con un po' di fantasia, si possono fare corse spericolate, sulle piste più diverse e mozzafiato. E se invece di una pista si disegna un labirinto, eccoci subito alla ricerca della via d'uscita, attenti a non toccare i muri che celano insidiosi trabocchetti...

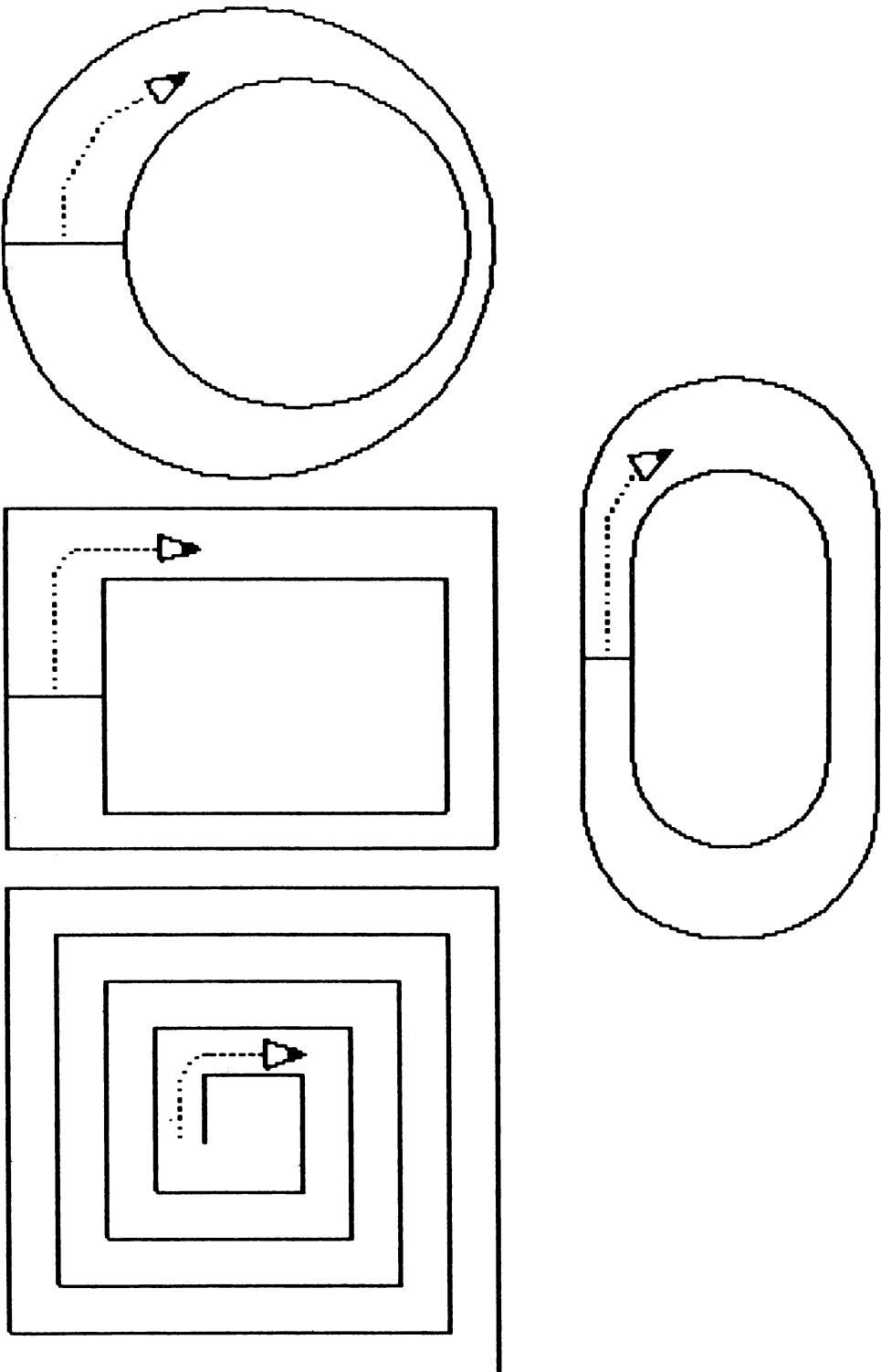
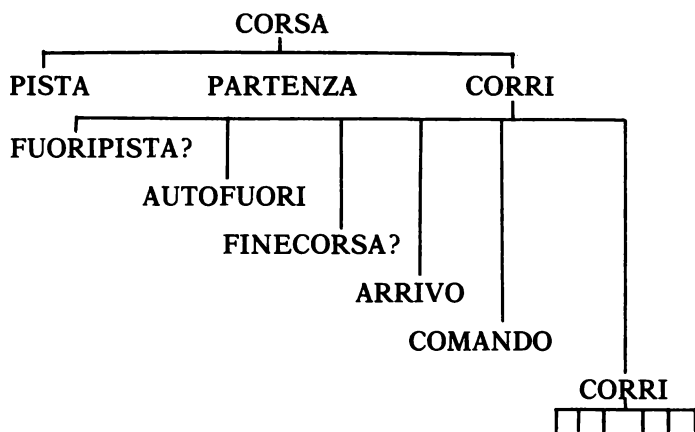



Figura 12.3 Piste per la tartaruga

SUGGERIMENTO

Il gioco della corsa in pista presentato qui, anche se molto semplice, riesce a divertire molta gente. In più, la libertà di tracciare qualunque pista venga in mente e di guidarvi la tartaruga "a mano libera" offre eccellenti occasioni di praticare le possibilità di movimento e di costruzione grafica della tartaruga.
La procedura PILOTA va bene anche per costruire semplici disegni come quelli del Capitolo 6, e torna utile quindi per arricchire le prime esperienze grafiche con la tartaruga. Può darsi inoltre che la PILOTA susciti interesse per come è scritta, e sia un esempio di tecnica recursiva e di comandi condizionali.
Il seguito di questo capitolo riguarda un'attività piuttosto impegnativa di elaborazione dati, che può rivelarsi troppo pesante per molti principianti: in questo caso, è meglio non insistere e continuare ancora per un po' con la corsa in pista più semplice.

12.4 In corsa — Parte II

Vediamo adesso come si può far controllare al computer la corsa della tartaruga: una versione semplice del gioco CORSA è riportata in Appendice A. Per pilotare la tartaruga, si usano i tasti V, L, D e S.
Qui si vede come funziona CORSA. Nel prossimo paragrafo, si considerano alcuni modi per migliorarla.
Mentre la tartaruga corre lungo la pista circolare, il computer controlla se esce di pista o se raggiunge indenne la linea del traguardo. A gara terminata, si vede anche il punteggio, cioè il tempo impiegato a fare il giro. Il gioco consiste nel fare un giro completo nel tempo più breve, senza però uscire mai di pista. Una procedura chiamata RIPARTI consente inoltre di ricominciare la corsa, senza però dover ridisegnare la pista.
In CORSA si impiegano tre procedure di servizio, già usate in TIRO.A.SEGNO (descritta nel Capitolo 10): CERCHIETTO, DISTANZA E LEGGITASTO (si possono caricare in memoria dal disco dopo averle copiate dall'Appendice A). Ecco lo schema di connessione del programma:

La superprocedura CORSA è semplice:

```
PER CORSA
PISTA
PARTENZA
CORRI 0
FINE
```

Il gioco è preparato da PISTA e PARTENZA (le vedremo poi) ed è condotto da CORRI:

```
PER CORRI :TEMPO
SE FUORIPISTA? AUTOFUORI STOP
SE FINECORSA? ARRIVO STOP
AVANTI :META
COMANDO
CORRI :TEMPO+1
FINE
```

Con la variabile :TEMPO si conteggia il tempo trascorso in gara. Il computer Commodore non misura direttamente il tempo: facciamo qui allora una misura indiretta, incrementando di uno l'input di ogni nuova CORRI. Le procedure più interessanti sono FUORIPISTA? e FINECORSA? che verificano la posizione della tartaruga e decidono se la corsa è finita. FUORIPISTA? ha in output la parola "VERO se la tartaruga oltrepassa il bordo della pista: se la tartaruga rimane all'interno, l'output di FUORIPISTA? è la parola "FALSO. Dato che la pista è circolare, con circonferenza esterna di raggio 70 e circonferenza interna di raggio 50, la procedura passa in output la parola "VERO se la tartaruga dista dal centro più di 70 o meno di 50. Ecco le istruzioni:

```
PER FUORIPISTA?
SE (DISTANZA 0 0)>70 RIPORTA "VERO
SE (DISTANZA 0 0)<50 RIPORTA "VERO
RIPORTA "FALSO
FINE
```

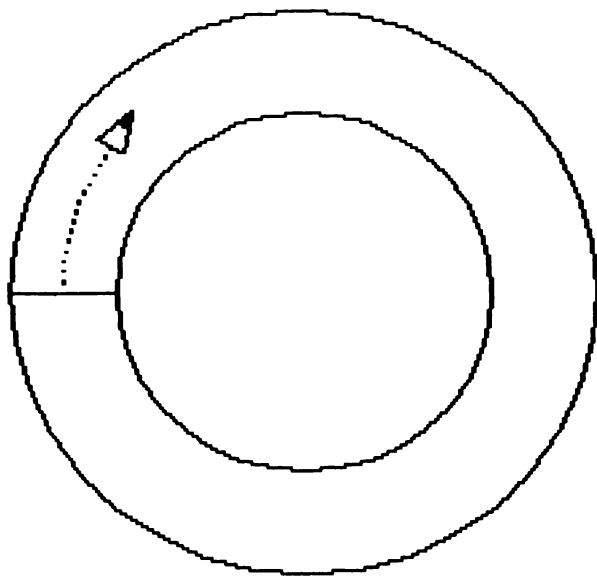
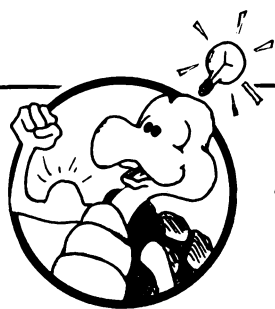



Figura 12.4 La tartaruga dopo la partenza

Conviene scegliere una pista facile da controllare. Nel nostro esempio è stata scelta la forma circolare perché il controllo dell'uscita di pista è veramente semplice: basta vedere se la distanza della tartaruga dal centro è

maggiore del raggio esterno oppure minore di quello interno. La scelta di una forma facile per la pista risparmia un sacco di grattacapi e ci fa ottenere un gioco che funziona come si deve, senza troppa fatica.

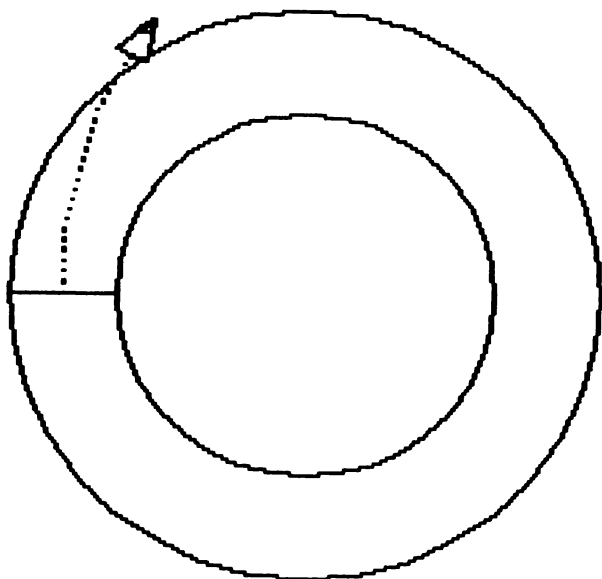



Figura 12.5 Ecco la tartaruga fuori pista

Tornando alla CORRI, se l'output di FUORIPISTA? è "VERO, allora viene eseguito il resto della prima istruzione dove AUTOFUORI scrive un messaggio, e CORRI si arresta:

```
PER AUTOFUORI
STAMPA [SEI USCITO DI PISTA!]
FINE
```

Se invece l'output di FUORIPISTA? è "FALSO, si passa all'istruzione successiva, dove si controlla l'output di FINECORSA?. Qui le cose sono un po' più complicate: la linea di arrivo è disposta in orizzontale, all'altezza Y=0. Quando la tartaruga la attraversa, si sposta da un punto con coordinata Y negativa ad uno con coordinata Y positiva, come si vede in Figura 12.6. Bene, la FINECORSA? si basa proprio su questa osservazione per decidere se la tartaruga è arrivata al traguardo.
Quano la tartaruga taglia il traguardo, si avverano due fatti: la sua coordinata Y attuale è maggiore di zero, e la coordinata Y precedente è mino-

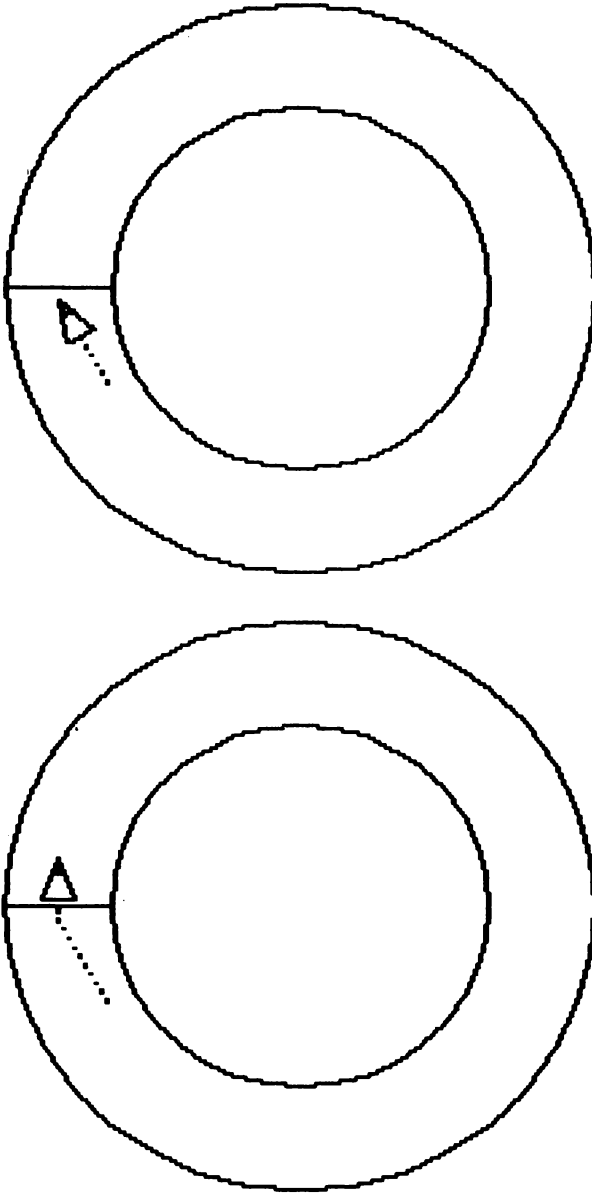



Figura 12.6 La tartaruga taglia il traguardo

re di zero. Allora il computer deve confrontare il punto in cui si trova la tartaruga con il punto da cui proviene, per risolvere il problema di FINE-CORSA?. Per sapere dove si trova la tartaruga, si usa il comando YCOR

del Logo. Per sapere da dove proviene, si impiega invece una variabile, chiamata qui "YVEC (ci si può limitare alla sola coordinata Y, senza badare alla X).
Ecco FINECORSA?:

```
PER FINECORSA?
SE TUTTIVERI? (YCOR>0) (:YVEC<0) RIPORTA "VERO
ASSEGNA "YVEC YCOR
RIPORTA "FALSO
FINE
```

La prima istruzione controlla se la condizione di arresto si è avverata. Il comando Logo TUTTIVERI? controlla se valgono "VERO tutti e due i suoi dati di input (YCOR>0) e (:YVEC<0): se così è, FINECORSA? passa in output la parola "VERO. Se così non è, si assegna alla coordinata Y attuale (data da YCOR) il nome "YVEC, e si passa in output la parola "FALSO. Quando FINECORSA? ha in output la parola "VERO, viene richiamata la ARRIVO, che scrive un messaggio e arresta la corsa:

```
PER ARRIVO
STAMPA [SEI ARRIVATO AL TRAGUARDO]
STAMPA FRASE [CON IL TEMPO DI] :TEMPO
FINE
```

TRABOCCHETTO

Attenti a capire bene la funzione del comando STOP alla fine della prima e della seconda istruzione di CORRI. Esso serve per arrestare la procedura CORRI, quando FINECORSA? o FUORIPISTA? assumono il valore "VERO. Se si elimina lo STOP finale, o lo si scrive all'interno delle procedure FINECORSA? e FUORIPISTA? non si ottiene lo stesso effetto, perché così la CORRI non si arresta. Ricordate sempre che STOP agisce solo sulla procedura a cui appartiene.
Per far fermare CORRI, si può scrivere il comando PUNTOACAPO alla fine di FINECORSA? e FUORIPISTA?. PUNTOACAPO è un comando un po' drastico (quasi come un freno di emergenza!), che fa arrestare tutte le

procedure. Attenzione però ad usarlo solo quando non se ne può proprio fare a meno: in molti casi, rende più complicata l'estensione del programma.

La penultima procedura di CORRI è COMANDO, che non cambia rispetto a prima:

```
PER COMANDO
ASSEGNA "COMAN LEGGITASTO
SE :COMAN = " STOP
SE :COMAN = "V ASSEGNA "META :META+5 STOP
SE :COMAN = "L ASSEGNA "META :META-5 STOP
SE :COMAN = "D DESTRA 30 STOP
SE :COMAN = "S SINISTRA 30 STOP
FINE
```

Nell'ultima istruzione di CORRI, viene chiamata un'altra CORRI, che riprende il percorso con il tempo aumentato di un'unità.
Vediamo ora le due procedure di apertura, in CORRI, cioè PISTA e PARTENZA:

```
PER PISTA
DISEGNA
NASTARTA
CERCHIETTO 50
CERCHIETTO 70
SINISTRA 90
SULAPENNA AVANTI 50
GIULAPENNA AVANTI 20
SULAPENNA INDIETRO 70
DESTRA 90
FINE
PER PARTENZA
VAXY (-60) 0
ASDIR 0
AVANTI 1
MOSTARTA
ASSEGNA "YVEC 1
ASSEGNA "META 0
FINE
```

Non è difficile seguire il funzionamento di queste due procedure. Le due CERCHIETTO richiamate in PISTA tracciano due circoli intorno al cen-

tro dello schermo. PARTENZA porta la tartaruga alla linea di partenza, assegna per "YVEC un valore maggiore di zero e fissa per la tartaruga una velocità iniziale nulla, con ASSEGNA "META 0.

SUGGERIMENTO

Realizzare questo gioco sembra semplice, ma non lo è. Qui si è presentato un caso particolare, con la pista ad anello che rende facile il controllo del percorso e la linea del traguardo orizzontale, che non crea grossi problemi per la condizione di arresto all'arrivo.
Appena però si cambia forma alla pista, incominciano i problemi. In teoria, è sempre possibile far controllare al computer se la tartaruga va fuori pista, ma in pratica occorre risolvere problemi matematici piuttosto ardui. Di fatto, ci si deve accontentare di piste dalla forma semplice (e anche queste sono affrontate con successo solo da studenti delle scuole superiori). Più avanti si vedranno alcuni esempi significativi.
C'è poi il fatto che il computer, eseguendo calcoli più complicati, rallenta il moto della tartaruga, rischiando di guastare gran parte del divertimento. È vero, molti giochi di animazione da computer sono complicati e, nello stesso tempo, veloci: la loro programmazione è però fatta in linguaggio macchina (illeggibile o quasi, a vista) da programmatori veramente esperti.
In effetti, qui si vede che il Logo, per essere così chiaro, paga lo scotto di una minor velocità di esecuzione: ecco perché bisogna fare attenzione, in casi come il nostro, a non esagerare con i calcoli.
Anche in questo caso, molta parte del successo di chi impara dipende dal discernimento di chi insegna, che dovrebbe essere tanto bravo da presentare all'allievo i problemi che questo è in grado di affrontare. Troppo spesso, invece, si vede che un piccolo si scoraggia perché proprio non ce la fa a risolvere un problema ancora troppo grande per lui (come ad esempio il controllo della tartaruga su una pista un po' complessa). Insomma, è importante avere il senso della misura: se si riesce a far capire che spesso per risolvere un problema conviene semplificarlo, rimandando certe difficoltà a un momento successivo, si è ottenuto molto, e non solo dal punto di vista dell'apprendimento del Logo!
Se si ha un allievo che vuole assolutamente provare una pista complicata, lasciatelo fare, naturalmente, ma senza fargli affrontare il problema del controllo del percorso con il computer. Dopo che l'esplorazione delle nuove forme si sarà sfogata, proponete un caso più semplice, da controllare con il computer.

12.5 Variazioni sulla corsa della tartaruga

Vediamo ora come si può rendere più interessante la corsa in pista: come al solito, le indicazioni date qui valgono solo a titolo dimostrativo e senz'altro il lettore troverà da solo molte estensioni da fare a CORSA. Inoltre, le estensioni non vengono presentate in tutti i dettagli, ma solo descritte in generale, come per le modifiche a TIRO.A.SEGNO viste nel Capitolo 10. Ecco dunque alcune idee:

1. Fare una presentazione del gioco. Basta decidere che scritte far comparire, e aggiungere una procedura PRESENTAZIONE in CORSA. Ricordate però che la presentazione serve solo a chi è nuovo del gioco, mentre chi già lo conosce la trova ben presto noiosa: lasciate quindi la scelta a chi gioca.
2. Modificare le scritte di ARRIVO e AUTOFUORI, per renderle più interessanti.
3. Memorizzare il record di velocità sul percorso (cioè, il tempo più breve impiegato per fare il giro). Ad ogni nuova gara, si può anche far comparire il record valido al momento. Ci vuole una nuova variabile, chiamata ad esempio "RECORD, e una nuova procedura, da richiamare in ARRIVO:

```
PER PARAGONE
SE (:TEMPO<:RECORD) ASSEGNA "RECORD :TEMPO
STAMPA FRASE [IL RECORD È ORA] :RECORD
FINE
```

   Bisogna anche dare un valore iniziale a "RECORD, la prima volta che si gioca alla corsa in pista. Si può fare con una procedura scritta al momento:

```
PER PRIMORECORD
ASSEGNA "RECORD 500
CONSERVA "CORSA
CANCELLA PRIMORECORD
FINE
```

   Conviene scegliere un numero grande, così che al primo giro completo il record viene subito aggiornato. Come si vede, la procedura registra su disco il file "CORSA, che ora comprende il valore assegnato a "RECORD, e poi si cancella da sola: la PRIMORECORD non è infatti una procedura di "CORSA, e serve solo una volta, la prima. Se si vuole conservare il nuovo valore di "RECORD, bisogna poi registrare ogni volta "CORSA, prima di spegnere il computer o di passare ad altro, se no

quando lo si riprende si parte con il vecchio record.
4. Far comparire qualcosa di vivace quando, ad esempio, la tartaruga va fuori pista o taglia il traguardo. Si può aggiungere una procedura ESPLOSIONE in AUTOFUORI, e una BANDIERASU in ARRIVO. Attenzione però: se l'esplosione è troppo entusiasmante, nessuno vorrà più arrivare al traguardo!
5. Fare una pista con larghezza variabile. Se la pista è stretta, il gioco è più impegnativo. Si può presentare al giocatore la scelta del livello di difficoltà (facile, medio, difficile) proprio come nel gioco TIRO.A.SEGNO. Si può anche far scegliere direttamente la larghezza della pista, usando poi il comando di assegnazione ASSEGNA "LARGH LEGGINUMERO (la LEGGINUMERO è una procedura di servizio, riportata in Appendice A).
   Se si cambia la larghezza della pista, va cambiato anche il controllo in FUORIPISTA?, come anche la PARTENZA, per far partire la tartaruga dal centro della pista.
6. Modificare la forma della pista. Ora però diventa difficile controllare il percorso col computer. Qui si propongono alcune modifiche non troppo complicate (oltre a PISTA, bisogna modificare anche PARTENZA, per far partire la tartaruga dal centro della pista).

- Variazione 1. Due cerchi non concentrici.

Si può ottenere ciò cambiando in PISTA la posizione dei due cerchi.

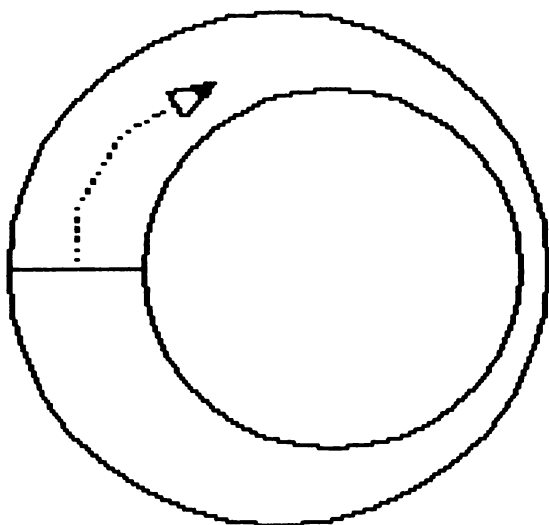



Figura 12.7 Pista a larghezza variabile

Bisogna poi cambiare FUORIPISTA? così che il LOGO controlli la distanza dal centro di ogni cerchio. Invece di DISTANZA 0 0 si usano le coordinate X e Y del centro di ogni cerchio come input di DISTANZA.

- Variazione 2. Due rettangoli (magari non concentrici).

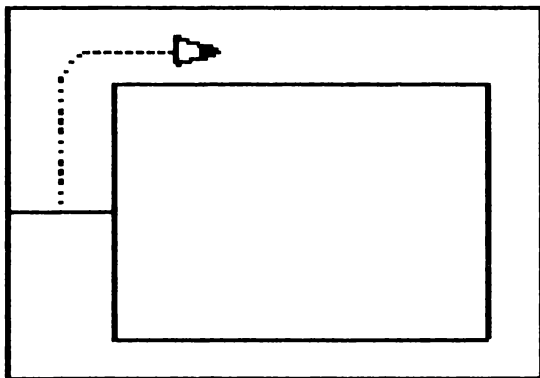



Figura 12.8 Pista rettangolare

Per tracciare la pista rettangolare, si possono impiegare le procedure di servizio TRETTAN, DENTRO? e FUORI? (son riportate in Appendice A). Queste procedure richiedono in input le coordinate X e Y di due vertici opposti del rettangolo che interessa, come si vede in Figura 12.9. Le coordinate del rettangolo interno sono chiamate XA,YA e XB,YB mentre quelle del rettangolo esterno sono XC,YC e XD,YD. Ci vuole una nuova

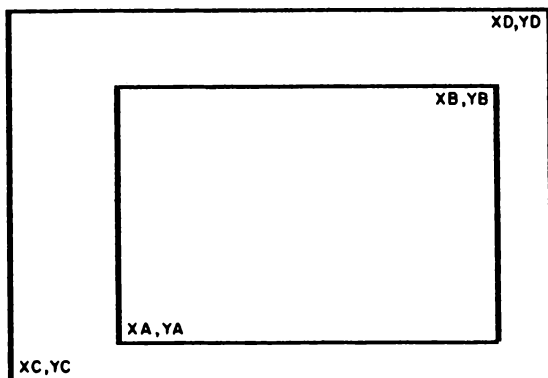



Figura 12.9 Definizione della pista rettangolare

versione di PISTA, per creare le nuove otto variabili e usarle con le procedure di servizio appena viste. Poi, anche DENTRO? e FUORI? fanno uso delle stesse variabili, nella procedura FUORIPISTA?.
Ecco le istruzioni:

```
PER PISTA
ASSEGNA "XA (-60) ASSEGNA "YA (-40)
ASSEGNA "XB 80 ASSEGNA "YB 60
ASSEGNA "XC (-100) ASSEGNA "YC (-65)
ASSEGNA "XD 100 ASSEGNA "YD 100
TRETTAN :XA :YA :XB :YB
TRETTAN :XC :YC :XD :YD
VAXY :XA 0
GIULAPENNA VAX :XC
FINE
PER FUORIPISTA?
SE DENTRO? :XA :YA :XB :YB RIPORTA "VERO
SE FUORI? :XC :YC :XD :YD RIPORTA "VERO
RIPORTA "FALSO
FINE
```

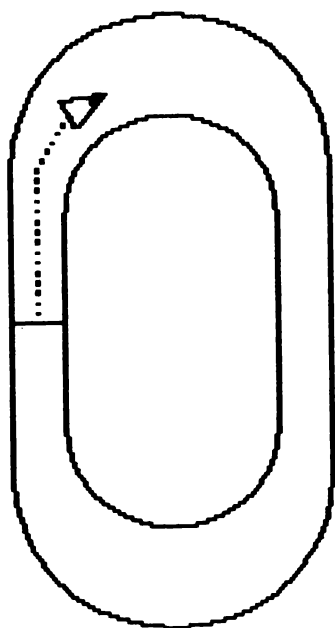



Figura 12.10 Pista ovale

La prima istruzione di PARTENZA deve inoltre cambiare così:

SULAPENNA VAXY ((:XA+:XC)/2) 0

Le altre procedure rimangono come sono.
Ecco ancora un'altra idea:

- Variazione 3. Pista di forma ovale

Quando si padroneggiano le piste circolari e quelle rettangolari, ci si può cimentare con la forma di Figura 12.10, che risulta dalla combinazione di un rettangolo e di un semicerchio. In bocca al lupo.

SUGGERIMENTO

Questi pochi esempi sono un'ottima occasione per presentare la geometria analitica in modo elementare. Ci vuole però molta pazienza, perché bisogna spiegare le coordinate X e Y, i numeri positivi e negativi, e le relazioni di confronto (come "maggiore " e "minore di") applicate ai numeri negativi.
Questo capitolo non intende certo presentare i concetti di base della geometria analitica, mà solo qualche esempio di animazione, che necessariamente richiede alcune conoscenze di analisi: queste si possono ottenere con un buon libro di algebra o, meglio ancora, con le spiegazioni di un buon insegnante di matematica. Se un piccolo allievo dimostra interesse all'argomento, non scoraggiatelo. Si accorgerà da solo della difficoltà, e dopo un po' di esplorazione lascerà perdere, magari per riprendere l'argomento dopo un anno o due, quando sarà più agguerrito.
Se poi non si riesce proprio a farne nulla, pazienza. Bisogna tener presente che non tutti possono fare tutto, e questo libro non fa eccezione.

13

Tempo di poesia

Ecco alcuni versi di chi scrive:

Il tuo mare
stride alto un gabbiano
lontano dal molo
ricorda ancora

Ed ecco alcuni versi del computer:

Il fiume vuoto
un tramonto aleggia
sopra le scure gocce
gentile farfalla immobile

Indovinate di chi sono questi:

Un bosco frusciante
una lucciola bisbiglia
oltre il fresco ruscello
incerta notte nebbiosa

Avete mai scritto una poesia? Tutti lo fanno, almeno una volta. Bisogna avere uno schema, occorre trovare parole adatte e soprattutto ci vuole sentimento: un ricordo, un'emozione, una speranza...chissà.
E il computer come fa? Che cosa credete che avesse "in mente" quando

ha composto le ultime due poesie presentate qui sopra? In effetti, un computer che scrive poesie fa pensare.
Naturalmente, sappiamo che il computer segue semplicemente una procedura, la quale stabilisce di selezionare certi tipi di parola secondo uno schema fisso. Ed esso seleziona le parole da lunghe liste di nomi, verbi, aggettivi, ecc. in un modo che non si può certo considerare "creativo". Se mai, è creativo il programmatore che ha scritto la procedura e ha indicato lo schema da seguire, o chi ha preparato le liste di parole.
Ma uno scrittore non fa lo stesso? Segue un certo schema, per forza (anche la grammatica e la sintassi sono uno schema), anche se dispone di una scelta molto più vasta di schemi e di un repertorio di parole molto più ricco. Alcuni scrittori ricorrono persino a dizionari dei sinonimi, per trovare le parole adatte: in che cosa dunque il computer è davvero diverso? Supponiamo che abbia anch'esso una scelta di schemi molto vasta e un vocabolario ricchissimo: potrebbe allora scrivere come noi? No, questo no. Per noi ciò che scriviamo, e soprattutto la poesia, parla al sentimento, al ricordo, significa emozione: e questo, il computer non lo può certo avere, anche se un programmatore molto bravo può persino riuscire a programmare un computer a scrivere "poesie" che sembrano autentiche, e che possono magari ingannare dei veri esperti.
In questo capitolo si considerano alcuni modi possibili per esplorare con il Logo il mondo delle parole e delle frasi, così come abbiamo esplorato il mondo delle figure grafiche. Con ciò avremo anche l'occasione di imparare qualcosa intorno alla struttura del nostro linguaggio. Anzi, in questo capitolo ciò che conta non è tanto pensare al Logo, ma analizzare il nostro modo di esprimerci, la nostra lingua, così da poterne trasferire una parte almeno in un programma di computer. Pensare al computer come ad un "essere" vuol dire fare della fantascienza, anche se oggi qualcuno riesce a ottenere storie o poesie "sintetiche": semplicemente, il confine della fantascienza non è così netto come credevamo.

Non sono usati nuovi comandi
Vengono usati i file "POETA, "DIMMI e "LEGGINUMERO dell'Appendice A.

Nuove procedure di servizio usate:

Nome	**Esempio**
DIMMI	STAMPA DIMMI [CASA RAGAZZO GATTO]

13.1 Frasi sintetiche

Ora vediamo in che modo si possono combinare parole in frasi, con procedure scritte in Logo. Viene usato il file 'DIMMI che è anche riportato nell'Appendice A. Ecco qui alcuni esempi:

Il topo corre
Una ragazza salta
Il computer elabora

Come si può notare, viene seguito uno schema fisso: articolo, nome, verbo. Gli articoli sono parole che aiutano a individuare i nomi, e sono ad esempio: un, uno, una, il, lo, la, i, gli, le. I nomi indicano un oggetto del nostro linguaggio, cioè una cosa, una persona, un luogo, come ad esempio: casa, ponte, libro, giocatore, bambino, sciocchezza, topo. I verbi indicano invece azioni, come ad esempio: vola, corre, pensa, parla, elabora. Ecco un programma per raggruppare parole a caso in semplici frasi:

```
PER PRIMAFRASE
RIPORTA (FRASE ARTICOLO NOME VERBO)
FINE
```

Quando si usa il comando FRASE con più di due dati di input (come in PRIMAFRASE) bisogna ricordarsi di mettere fra parentesi il comando stesso, con tutti i suoi dati. Se non si usano le parentesi, FRASE può avere solo due dati di input.
Il comando FRASE combina parole e liste per formare una lista più lunga, come si è visto nel Capitolo 9. ARTICOLO, NOME e VERBO sono procedure Logo, riportate qui di seguito, che scelgono a caso parole comprese in apposite liste:

```
PER ARTICOLO
RIPORTA DIMMI :ARTICOLI
FINE

PER NOME
RIPORTA DIMMI :NOMI
FINE

PER VERBO
RIPORTA DIMMI :VERBI
FINE
```

La DIMMI è una procedura di servizio che fornisce in output un elemento della sua lista di input. Ecco qui alcune liste:

```
ASSEGNA "ARTICOLI [IL LO LA I GLI LE UN UNO UNA]
ASSEGNA "NOMI [RAGAZZA FRIGO COMPUTER TOPO CANGURO
    CASA BICI BIMBO]
ASSEGNA "VERBI [VA CORRE BACIA SALTA VOLA GIOCA
    PRENDE NUOTA DORME]
```

Proviamo:

```
STAMPA PRIMAFRASE
UN COMPUTER NUOTA
```

Proviamo ancora:

```
STAMPA PRIMAFRASE
IL FRIGO VOLA
```

Eh sì, ci vuole altro per fare frasi sensate. Quelle ottenute sono frasi possibili, ma completamente sconclusionate. Vedremo poi come migliorare il senso. Per ora andiamo avanti così.
Ecco altri tentativi:

```
RIPETI 5[STAMPA PRIMAFRASE]
UN BIMBO VOLA
LA BICI CORRE
UN CANGURO BACIA
IL COMPUTER BACIA
LA BICI SALTA
```

Ora aggiungiamo qualche nuovo elemento, senza preoccuparci ancora del senso delle frasi:

```
PER AGGETTIVO
RIPORTA DIMMI :AGGETTIVI
FINE

PER AVVERBIO
RIPORTA DIMMI :AVVERBI
FINE
```

Ecco le liste associate (ricordiamo che gli aggettivi descrivono qualità dei nomi, come caldo, alto, bravo; gli avverbi sono invece parole che modificano altre parole, in quantità, qualità, ecc. come ad esempio adagio, tristemente, felicemente, lentamente):

```
ASSEGNA "AGGETTIVI [CARINO CALMO RICCO POVERO LIETO
   TRISTE ARRABBIATO BELLO RAPIDO LENTO]
ASSEGNA "AVVERBI [RAPIDAMENTE LENTAMENTE LIETAMENTE
   TRISTEMENTE PRECISAMENTE QUIETAMENTE
   DOLCEMENTE PIACEVOLMENTE]
```

Potete naturalmente aggiungere a queste liste le parole che volete (e che vi piaceranno certo di più).
Passiamo ora allo schema delle frasi: si possono provare schemi sensati e schemi assurdi, con un po' di varietà, giusto per fare pratica. Con i cinque tipi di parole appena visti, si possono costruire moltissimi schemi diversi (è naturalmente possibile ripetere più volte, in uno schema, lo stesso tipo di parola, e si possono anche unire più schemi). Ecco qui alcune idee, tanto per incominciare:

```
PER SECONDAFRASE
RIPORTA (FRASE ARTICOLO AGGETTIVO NOME VERBO
   AVVERBIO)
FINE
STAMPA SECONDAFRASE
IL RICCO TOPO CORRE LIETAMENTE
```

Proviamo a invertire l'ordine dei tipi di parola:

```
PER TERZAFRASE
RIPORTA (FRASE AVVERBIO VERBO NOME AGGETTIVO
   ARTICOLO)
FINE
STAMPA TERZAFRASE
DOLCEMENTE DORME TOPO TRISTE UN
```

E se mettessimo i tipi di parola in ordine alfabetico?

```
PER QUARTAFRASE
RIPORTA (FRASE AGGETTIVO ARTICOLO AVVERBIO NOME
   VERBO)
FINE
STAMPA QUARTAFRASE
POVERO IL TRISTEMENTE CANGURO VA
```

Proviamo a caso:

```
PER QUINTAFRASE
RIPORTA (FRASE ARTICOLO AVVERBIO NOME VERBO
```

```
AGGETTIVO)
FINE
STAMPA QUINTAFRASE
UN LENTAMENTE FRIGO VA TRISTE
```

Un tipo di parola può comparire più d'una volta:

```
PER SESTAFRASE
RIPORTA (FRASE AVVERBIO ARTICOLO AGGETTIVO AGGETTIVO
   NOME VERBO)
FINE
STAMPA SESTAFRASE
DOLCEMENTE UN CARINO CALMO BIMBO DORME
```

Perché non combiniamo due frasi? Magari la seconda e la terza, che sono una l'inverso dell'altra:

```
PER SETTIMAFRASE
RIPORTA (FRASE SECONDAFRASE [E] TERZAFRASE)
FINE
STAMPA SETTIMAFRASE
IL RAPIDO BIMBO BACIA PRECISAMENTE E DOLCEMENTE
   PRENDE FRIGO POVERO UNO
```

Supponiamo ora di avere dodici schemi diversi: qui di seguito è indicata una procedura che sceglie a caso lo schema da applicare di volta in volta (con il comando ACASO 12 si estrae un numero fra 0 e 11, per cui nell'istruzione di estrazione viene aggiunto uno):

```
PER FRASI
ASSEGNA "NUM (1+ACASO 12)
SE :NUM = 1 STAMPA FRASE1
SE :NUMERO = 2 STAMPA FRASE2
...
...
SE :NUM = 12 STAMPA FRASE12
FRASI
FINE
STAMPA FRASI
LENTAMENTE SALTA RAGAZZA TRISTE UN
UN LIETO FRIGO GIOCA PIACEVOLMENTE
IL RICCO TOPO BACIA PRECISAMENTE
LENTAMENTE IL RAPIDO LENTO CANGURO DORME
IL CARINO FRIGO VOLA LIETAMENTE
```

```
UN COMPUTER VA
IL POVERO TOPO SALTA DOLCEMENTE
IL PRECISAMENTE RAGAZZA POVERO NUOTA
IL RICCO FRIGO VA LIETAMENTE E QUIETAMENTE VOLA
   RAGAZZA RICCO UN
CALMO DOLCEMENTE UN CANGURO VA
...
...
```

Se una frase proprio non piace, se ne può scartare lo schema, sostituendolo con un altro. Provate a vedere quali frasi rispettano le regole della lingua italiana e quali invece no.

13.2 Frasi sensate

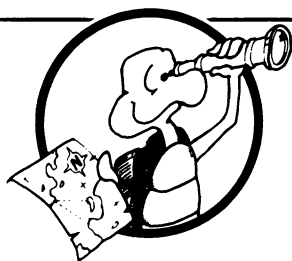
È possibile costruire frasi più sensate? Qualcosa si può fare: ad esempio, si possono selezionare parole (nomi, aggettivi, verbi ...) che abbiano fra di loro una vaga corrispondenza, un'assonanza magari, o che si riferiscano allo stesso argomento (lo sport, la natura, il gioco, la fantascienza, e così via). Con liste di parole collegate, anche le frasi sintetiche guadagnano in senso. Ecco qui un estratto delle liste usate nel programma POETA:

```
ASSEGNA "NOMI [CASCATA FIUME BREZZA LUNA PIOGGIA
   VENTO MARE MATTINO NEVE LAGO TRAMONTO OMBRA
   PINO FOGLIA LUCCICHIO ALBA BOSCO]
ASSEGNA "VERBI [SCUOTE PORTA [SI NASCONDE] DORME
   [SI ALLUNGA] MORMORA VOLA VIBRA CADE]
ASSEGNA "AGGETTIVI [AUTUNNALE NASCOSTO FRIZZANTE
   RIBOLLENTE VORTICOSO VERDE AMARO NEBBIOSO
   SILENZIOSO VUOTO]
ASSEGNA "AVVERBI [RAPIDAMENTE LENTAMENTE
   ALLEGRAMENTE TRISTEMENTE QUIETAMENTE
   FORTEMENTE DOLCEMENTE]
ASSEGNA "ARTICOLI [UN IL UNO UNA LA LO]
```

Come si vede, alcuni elementi di una lista possono a loro volta essere brevi liste (come [SI NASCONDE]).
Proviamo ancora con le nuove liste:

```
FRASI
FORTEMENTE IL VORTICOSO NEBBIOSO VORTICOSO MARE
```

DORME
DOLCEMENTE IL FRIZZANTE VUOTO NASCOSTO BOSCO
MORMORA
LENTAMENTE PORTA FIUME AUTUNNALE LO
UNO QUIETAMENTE PINO VERDE SI ALLUNGA
ALLEGRAMENTE PORTA MATTINO RIBOLLENTE UNO
UN AMARO LAGO SCUOTE RAPIDAMENTE
UN AMARO TRAMONTO PORTA QUIETAMENTE E LENTAMENTE
CADE VENTO AMARO UN
QUIETAMENTE IL NASCOSTO NASCOSTO MARE VIBRA
VORTICOSO RAPIDAMENTE IL BOSCO PORTA
LO FRIZZANTE OMBRA DORME ALLEGRAMENTE

Gli schemi usati sono i soliti, ma ora il senso è un poco migliorato. Visto quanto è importante una buona scelta di parole?

ESPLORAZIONE

Provate a raccogliere liste di parole che vi sembrano interessanti: lo si può fare in due, perché in casi come questo due teste sono veramente meglio di una sola. Va bene qualunque argomento: politica, astronomia, sport, cinema, computer, ecc.

13.3 POETA

Nel programma POETA le frasi sono composte con un repertorio di parole scelte con cura e disposte in schemi che favoriscono un senso compiuto. Le procedure complete sono registrate nel file "POETA riportato nell'Appendice A.
Ecco la procedura principale:

```
PER POETA
STAMPA VERSO1
STAMPA VERSO2
STAMPA VERSO3
```

Ecco un esempio:

```
POETA
UN LIMPIDO STAGNO
LA RONDINE SFRECCIA SOTTO UN ABETE SILENZIOSO
SELVAGGIA VERDE LUNA
```

Si indovina lo schema?

```
PER VERSO1
RIPORTA (FRASE ARTICOLO AGGETTIVO NOME)
FINE
PER VERSO2
RIPORTA (FRASE ARTICOLO NOME VERBO PREPOSIZIONE
   ARTICOLO NOME AGGETTIVO)
FINE
PER VERSO3
RIPORTA (FRASE AGGETTIVO AGGETTIVO NOME)
FINE
```

Come si vede, c'è un nuovo tipo di parola, a cui si riferisce PREPOSIZIONE (le preposizioni indicano relazioni, ad esempio di luogo e tempo, fra nomi). Ecco la nuova procedura e la lista a cui ricorre:

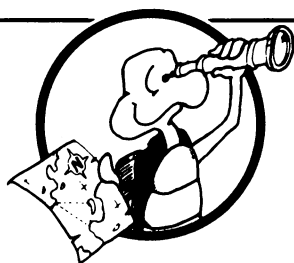
```
PER PREPOSIZIONE
RIPORTA DIMMI :PREPOSIZIONI
FINE
ASSEGNA "PREPOSIZIONI [SOPRA SOTTO FUORI DENTRO OLTRE
   PRESSO [VICINO A] [INTORNO A] CON PER FRA DI TRA IN DA]
```

Le liste dei nomi, dei verbi e degli aggettivi di POETA sono molto più ricche di quelle viste nel paragrafo 13.2: sono tutte riportate in Appendice A.
Nel file POETA si trova anche una procedura chiamata POESIE, che domanda quante poesie sintetiche si vogliono vedere, e le produce automaticamente:

```
PER POESIE
PULISCITESTO
```

```
STAMPA [QUANTE POESIE VUOI?]
ASSEGNA "N LEGGINUMERO
PULISCITESTO STAMPA [ ] STAMPA [ ]
STAMPA FRASE :N [LOGOPOESIE]
STAMPA [ ]
RIPETI :N [POETA STAMPA [ ]]
FINE
```

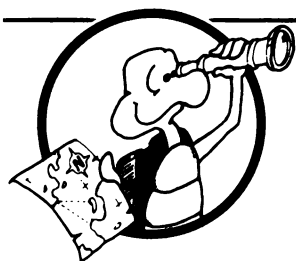
Viene usata la LEGGINUMERO, una procedura di servizio già incontrata e che si trova nel file "LEGGINUMERO riportato in Appendice A). Proviamo:

```
POESIE
QUANTE POESIE VUOI?
3

3 LOGOPOESIE

UNA RIBOLLENTE POLVERE
IL FIUME SI NASCONDE SOTTO LA VERDE NOTTE
AUTUNNALE SILENZIOSO CORVO

UN VUOTO MATTINO
IL CORVO MORMORA SOPRA LA SELVAGGIA NEVE
RIBOLLENTE FRIZZANTE OMBRA

LA AUTUNNALE CIVETTA
LA FOGLIA SI ALLUNGA SOPRA LA RIBOLLENTE SERA
LIMPIDO NASCOSTO MATTINO
```

ESPLORAZIONE

Provate a scrivere una procedura di poesia per conto vostro, con diversi schemi di composizione dei versi, che possono anche variare di numero. Il modo più semplice di modificare le liste di parole usate è battere EDITA NOMI: quando si arriva al termine delle modifiche si può lasciare l'editor con CTRL-C, come al solito.

A proposito, chi si può considerare l'autore dei versi prodotti da POETA? Il computer, il programmatore, oppure chi ha fornito il modello per gli schemi di composizione? Magari nessuno di questi, oppure invece tutti e tre. Chissà.

13.4 Alla scoperta del mondo del linguaggio

Qui si considerano ancora, a titolo di esempio, altre possibili applicazioni delle idee appena presentate.
Per scrivere altre procedure come POETA, si può benissimo prendere a modello una poesia vera, e magari analizzarla per scoprirne lo schema segreto (in molti casi, la ricerca è veramente stimolante). Invece di una poesia si può prendere anche una frase famosa, o magari un proverbio. Oltre allo schema del modello, si può conservare anche qualche parola, in modo da ottenere una maggiore somiglianza, che può a volte tornar comoda, anche per dare un certo senso alle frasi che si ottengono dal computer.

ESPLORAZIONE

Ecco qui ad esempio cosa si può fare prendendo a modello la celebre frase dell'*Amleto* di Shakespeare "Essere o non essere, questo è il dilemma":

```
PER PALLIDOPRENCE
ASSEGNA "VERB1 VERBO
RIPORTA (FRASE :VERB1 [O NON] :VERB1 [QUESTO È IL] NOME)
FINE
```

Attenzione, poi, che occorre una nuova lista di verbi, posti all'infinito (come volare, correre, ecc.).
E ora proviamo:

```
STAMPA PALLIDOPRENCE
DORMIRE O NON DORMIRE QUESTO È IL TRAMONTO
```

Nella prima istruzione della procedura, l'output di VERBO viene assegnato alla variabile "VERB1, perché così è possibile usare per due volte lo stesso verbo. Non si ottiene lo stesso effetto richiamando due volte la VERBO, perché questa estrae ogni volta un nuovo verbo, a caso:

```
PER PALLIPREN
RIPORTA (FRASE VERBO [O NON] VERBO [QUESTO È IL] NOME)
FINE
```

Proviamo ancora:

```
STAMPA PALLIPREN
MORMORARE O NON VOLARE QUESTO È IL LAGO
```

Prendete a modello una breve poesia, come ad esempio questi semplici versi:

La nebbia ha passi
di piccola gatta.
Viene e figge gli occhi innocenti
accovacciata in silenzio
sulle banchine del porto e sulle vie.
Ma poi va via.

Per prima cosa, bisogna scegliere le parole da tenere tali e quali, e quelle invece da estrarre a caso dalle liste. Poi si scrive la procedura:

```
PER NEBBIA
STAMPA (FRASE ARTICOLO NOME [HA PASSI])
STAMPA (FRASE [DI] AGGETTIVO ANIMALE)
STAMPA (FRASE VERBO [E] VERBO ARTICOLO NOME AGGETTIVO)
STAMPA (FRASE AGGETTIVO [IN] NOME)
STAMPA (FRASE [SULLE BANCHINE DEL] NOME [E SULLE VIE])
STAMPA (FRASE [MA POI] VERBO [VIA])
```

Occorre una nuova procedura ANIMALE, che estrae a caso nomi di animali da una lista. Ecco cosa si può ottenere:

```
NEBBIA
UN BOSCO HA PASSI
DI NASCOSTO CORVO
```

VA E SCUOTE IL MARE VERDE
AMARO IN POLVERE
SULLE BANCHINE DEL TRAMONTO E SULLE VIE
MA POI SI NASCONDE VIA

Ebbene, andiamo avanti fino in fondo, e senza farci troppi scrupoli di buon senso (in fondo, è un gioco): perché non prendiamo una novella, magari Cappuccetto Rosso? O Biancaneve? Come prima, si lasciano stare alcune parole e se ne fanno cambiare altre a caso dal computer:

CORVO CORVO DELLE MIE BRAME CHI È LA PIÙ VERDE DEL REAME?

Se gli esempi presentati vi sembrano un po' sciocchi, fatene voi di più seri (basta scegliere le parole da inserire in lista, e trovare gli schemi adatti). Ci vuole pazienza e fantasia, e si può far quasi tutto ciò che si vuole (se lo permette la memoria del computer).
Questo tipo di ricerca è addirittura l'oggetto del lavoro di specialisti di informatica, che impiegano computer molto potenti per sfornare poesie, romanzi, e altri "prodotti letterari" sintetici. Chi sa fino a quando saremo in grado di distinguere a colpo sicuro lo scritto di una persona dal prodotto di un computer? Forse un giorno potrete rispondere voi.

14

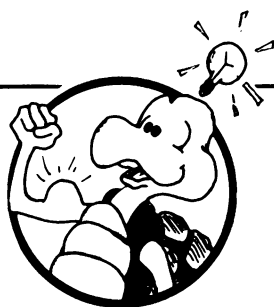
Funzionamento delle procedure di servizio

Tutto questo capitolo dovrebbe stare sotto l'insegna del SUGGERIMENTO, che tante volte compare nel resto del libro per indicare parti dedicate a chi già conosce il Logo e intende aiutare altri ad impararlo. Chi giunge qui dopo aver letto l'intero libro può comunque considerarsi a buon diritto un esperto del Logo, ed è senz'altro in grado di comprendere agevolmente la maggior parte di questo ultimo capitolo, anche perché molte procedure di servizio sono più semplici di certi programmi presentati negli ultimi capitoli.
Chi invece ricorre a questo capitolo durante la lettura del libro, per vedere come funziona una particolare procedura di servizio, se la può cavare se legge con molta attenzione la parte che lo riguarda. Attenzione però, che non tutto è facile, in questo capitolo.
Vi domanderete: perché distinguere dalle altre le procedure di servizio che, in fondo, non sono niente di speciale? Perché non vengono invece spiegate quando le si usa per la prima volta? Col nome di procedure di servizio vengono qui raccolte quelle procedure che facilitano l'esecuzione di altre, e che trovano impiego in più d'una circostanza. Con questi strumenti, poi, si costruiscono più facilmente le procedure che interessano. Del resto, che cosa rende tale un attrezzo, come ad esempio un martello, se non la nostra utilità? E non ci sognamo certo di imparare a costruire un martello, prima di usarlo. Così è anche per le procedure di servizio usate in questo libro.

Comando	**Abbreviazione**	**Esempi**
STATOSCHERMO	S	STAMPA STATOSCHERMO
UNOVERO?		SE UNOVERO? (XCOR>0) (YCOR>0) STOP
RADQ		STAMPA RADQ 100
TASTO?		SE TASTO? RIPORTA LEGGICAR
LEGGICAR	LC	ASSEGNA "TASTO LEGGICAR
NON		SE NON (XCOR>0) STOP
NUMERO?		SE (NON NUMERO? :R) RIPORTA LEGGINUMERO
CARATTERE		STAMPA CARATTERE 17

Sono usati i file "CERCHI, "CERCHIETTO, "RETTANGOLI, "DISTANZA, "LEGGITASTO, "DIMMI, "LEGGINUMERO e "ASPETTA dell'Appendice A.

Nuove procedure di servizio usate:

Nome	**Esempio**
QUALE	STAMPA QUALE 3[A B C D]

La possibilità di usare procedure già scritte da altri, come se fossero semplici comandi, rende il Logo ideale per l'apprendimento. Un insegnante può così usare il Logo per costruire su misura del singolo allievo un ambiente di procedure utilizzabili direttamente, come se fossero comandi nativi del Logo. Un insegnante che prepara per i suoi allievi procedure scritte in Logo costruisce in realtà un linguaggio di apprendimento, così come un artista che prepari procedure di disegno con la tartaruga costruisce un linguaggio di espressione grafica. Il linguaggio creato ad hoc dall'insegnante si fonde poi con quello che l'allievo viene costruendo da

sé, man mano che egli riesce a comprendere le procedure che all'inizio gli sono state presentate come "comandi nativi".

Prendiamo ad esempio le procedure per tracciare cerchi, presentate nel Capitolo 2 e usate prima ancora di parlar di procedure. Con alcuni insegnanti, gli allievi sono in grado di fare disegni che fanno uso di cerchi solo dopo aver imparato come si traccia un cerchio con la tartaruga e dopo aver visto come si scrive una procedura. Qui invece si segue una strada diversa: il principiante è in grado di far comparire un cerchio sin dall'inizio, prima ancora di immaginare l'esistenza di procedure con dati di input e di capire la geometria del cerchio: così subito si possono fare disegni interessanti e si ha l'impressione di poter comandare al computer. C'è tempo, poi, per le finezze di programmazione e la geometria del cerchio.
Le procedure di servizio si giustificano anche per un altro motivo: molto spesso la stessa piccola procedura viene usata in diversi contesti, e conviene allora registrarla con un nome tutto suo in modo da averla già pronta quando occorre di nuovo. Man mano che si scrivono nuovi programmi, è normale che le parti più ricorrenti vengano messe in serbo, per formare una piccola raccolta privata di procedure di servizio.
Ma allora, perché mai spiegare queste procedure di servizio? Non è meglio usarle e basta, e se mai lasciare che ognuno si crei le proprie, quando ne ha bisogno?
Forse. Però esse sono una buona occasione (l'ultima...) per approfondire certi aspetti del Logo che meritano attenzione, specie da parte di chi vuole poi proseguire per proprio conto.

14.1 Archi e cerchi

Le procedure CERCHIODES, CERCHIOSIN, ARCODES e ARCOSIN sono state presentate nel Capitolo 2. Esse si basano su due procedure, chiamate RCP e LCP, ciascuna delle quali prevede in input un dato numerico, che indica il raggio del cerchio da tracciare (la RCP serve per cerchi in senso orario, la LCP per cerchi in senso antiorario).
I cerchi tracciati con procedure Logo sono in realtà poligoni. Le procedure qui descritte tracciano poligoni di 36 lati, che riescono ad approssimare il cerchio abbastanza bene: RCP e LCP ruotano la tartaruga di 10 gradi, in modo che si ottenga il cerchio con 36 ripetizioni (compiendo un giro completo, cioè un arco di 360 gradi):

```
PER RCP :R
DESTRA 5
AVANTI :R*3.14159/18
DESTRA 5
FINE
```

Da dove viene la formula :R*3.14159/18? E perché si usano due istruzioni DESTRA 5 invece di una sola DESTRA 10?
Il numero 3.14159 è il valore approssimato della costante geometrica π (pi greco), che indica il rapporto fra la circonferenza del cerchio e il suo diametro. In Figura 14.1 la circonferenza è C, il diametro D e il raggio R. La trentaseiesima parte della circonferenza è S: questo è il passo della tartaruga, per tracciare il poligono a 36 lati.

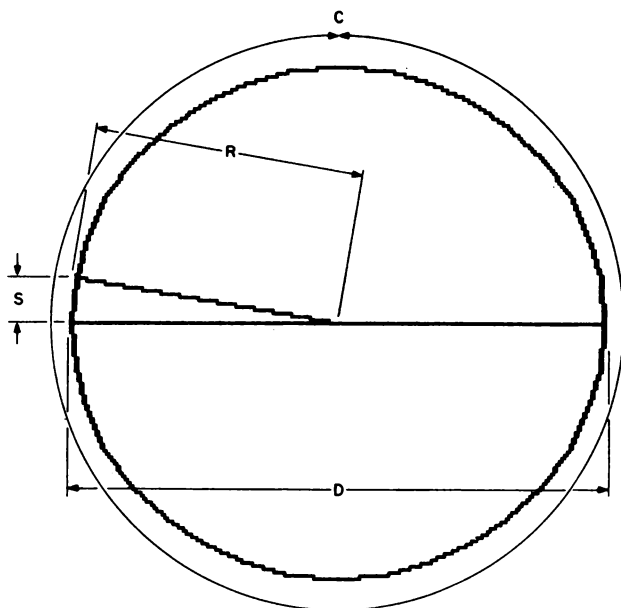



Figura 14.1 Ogni segmento S vale 10/360, cioè 1/36 dell'intera circonferenza

La relazione matematica fra la circonferenza e il suo diametro è:

$$C = 3.14159 \times D$$

Se si indica il doppio del raggio in luogo del diametro, si ha:

$$C = 3.14159 \times R \times 2$$

Per trovare quanto è lunga la trentaseiesima parte della circonferenza C, si dividono entrambi i membri dell'uguaglianza per 36:

S = C/36 = 3.14159×R×2/36

Sostituendo 2/36 con 1/18, si ottiene la formula che compare in RCP e in LCP:

S = 3.14159×R/18

In linguaggio Logo, l'ultima relazione diventa:

AVANTI :R*3.14159/18

Lasciando scritto :R*3.14159/18 invece di :R*0.17453 si ha un'istruzione più chiara, almeno per chi conosce l'uso di pi greco.
Vediamo ora perché RCP usa due istruzioni DESTRA 5 invece di una sola DESTRA 10: si usano due istruzioni perché così il cerchio tracciato ha il centro esattamente a 90 gradi rispetto all'orientamento iniziale della tartaruga. Usando invece una sola istruzione di rotazione, il centro del cerchio risulterebbe leggermente al di sopra (o al di sotto) del punto di partenza della tartaruga. Con un poligono di 36 lati, la differenza è appena percettibile: è molto chiara invece se il poligono ha meno lati, come un esagono. È facile vedere l'effetto diverso di questi due comandi:

RIPETI 6[AVANTI 50 DESTRA 60]

e

RIPETI 6[DESTRA 30 AVANTI 50 DESTRA 30]

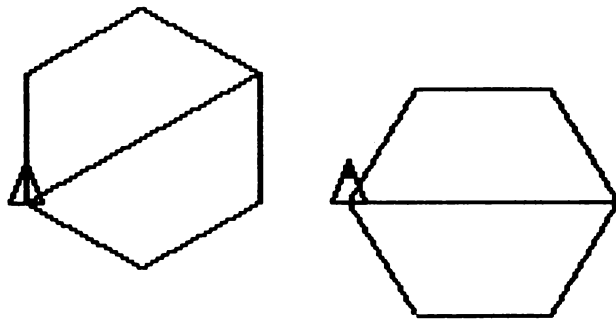



Figura 14.2 Con istruzioni di rotazione simmetriche rispetto all'istruzione di avanzamento, si ottiene un poligono perfettamente orizzontale

Se il poligono ha 36 lati, la differenza è molto minore, ma anche così sarebbe difficile realizzare per bene i disegni presentati in questo libro. Provate dunque a vedere la differenza fra queste altre istruzioni:

```
RIPETI 36[AVANTI 10 DESTRA 10]
```

e

```
RIPETI 36[DESTRA 5 AVANTI 10 DESTRA 5]
```

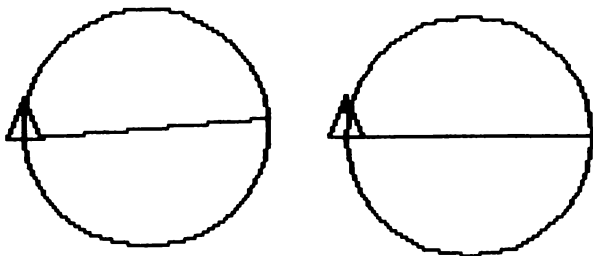



Figura 14.3 L'effetto di inclinazione è minore con poligoni di 36 lati

ARCODES e ARCOSIN tracciano archi di 90 gradi richiamando RCP e LCP per nove volte. Gli archi di 90 gradi (che si chiamano quadranti) servono in molte occasioni, ad esempio per arrotondare gli angoli di un rettangolo o per costruire semicerchi. Si possono ottenere archi di tutti i tipi, richiamando RCP e LCP un numero opportuno di volte.

14.2 CERCHIETTO

CERCHIETTO traccia un circolo intorno al punto in cui si trova la tartaruga. È utile in quei casi in cui il centro è già stabilito, ed è importante la distanza fra tale centro e il bordo del cerchio (ad esempio, nel gioco di tiro al bersaglio del Capitolo 10, o in quello della corsa in pista del Capitolo 13).
Anche CERCHIETTO usa RCP, richiamandola per 36 volte. Eccone la parte principale:

```
.....
NASTARTA
SULAPENNA AVANTI :R
DESTRA 90
GIULAPENNA RIPETI 36[RCP :R]
SINISTRA 90
SULAPENNA INDIETRO :R
```

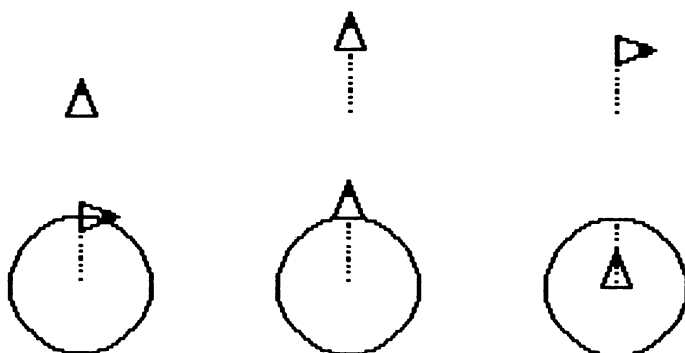



Figura 14.4 CERCHIETTO traccia un cerchio intorno alla tartaruga

Ecco un aspetto interessante di CERCHIETTO: la procedura lascia il pennino della tartaruga abbassato, quando termina, se così era all'inizio e, se la tartaruga era visibile all'inizio, lo diventa ancora alla fine. Questo effetto è ottenuto grazie a un comando non ancora visto, chiamato STATOSCHERMO. Provate ad esempio questo caso:

```
DISEGNA
STAMPA STATOSCHERMO
```

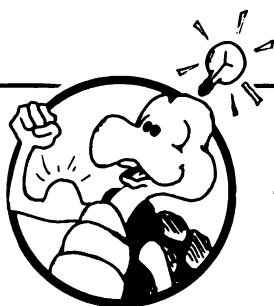
Il computer stampa:

```
VERO VERO 11 1 DISEGNA UNCOLORE DISCHERMO 14 6
```

Si ottiene, come si vede, una lista di nove elementi, alcuni dei quali parole ed altri numeri. Il primo elemento dice se il pennino è giù o su (VERO significa che il pennino è abbassato). Il secondo elemento si riferisce alla tartaruga (VERO significa che la tartaruga è visibile). Gli altri sette elementi non vengono usati in CERCHIETTO (per il loro significato, si può consultare il manuale del Logo Commodore 64). A questo punto potete seguire l'intera procedura CERCHIETTO:

```
PER CERCHIETTO :R
ASSEGNA "PEN? PRIMO STATOSCHERMO
ASSEGNA "VISTA? PRIMO MENPRI STATOSCHERMO
NASTARTA
SULAPENNA AVANTI :R
DESTRA 90
GIULAPENNA RIPETI 36[RCP :R]
SINISTRA 90
SULAPENNA INDIETRO :R
```

```
SE :PEN? GIULAPENNA
SE :VISTA? MOSTARTA
FINE
```

Nelle prime due istruzioni, si assegnano i nomi "PEN? e "VISTA? ai primi due elementi di STATOSCHERMO. Le due ultime istruzioni utilizzano queste due variabili: se :PEN? vale "VERO, il pennino viene abbassato, e se :VISTA? vale "VERO, viene mostrata la tartaruga. In caso contrario, il pennino è lasciato alzato e la tartaruga rimane nascosta.

IDEA

Con questo esempio, in cui la tartaruga ritorna, al termine di una procedura, al suo stato iniziale (non solo per quanto riguarda il punto in cui si trova, ma anche l'assetto del pennino e il modo di visualizzazione della tartaruga stessa) si mette in evidenza un accorgimento molto importante, che trova applicazione in moltissimi programmi: memorizzare le condizioni iniziali del sistema su cui si agisce, in modo da essere in grado di ripristinarle in qualunque momento (ad esempio, al termine di una particolare elaborazione).

14.3 Rettangoli

Le procedure di servizio per rettangoli sono realizzate per quei casi in cui interessa conoscere se la tartaruga è all'interno o all'esterno di un certo rettangolo (come nella variante della corsa con pista rettangolare del Capitolo 12). Vengono usati i comandi VAXY, VAX e VAY per disegnare un rettangolo, a partire dalle coordinate di due vertici opposti, che servono anche per determinare se la tartaruga si trova all'interno o all'esterno del rettangolo stesso. Queste procedure sono meno importanti

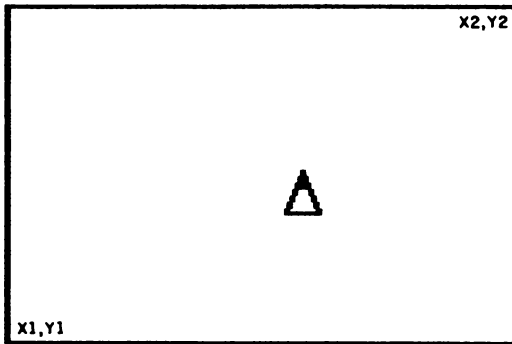



Figura 14.5 Rettangolo definito dalle coordinate di due vertici opposti

di quelle relative ai cerchi, ma offrono l'occasione per discutere meglio l'utilizzo delle coordinate cartesiane.
Come abbiamo visto, per usare le procedure TRETTAN, DENTRO? e FUORI? bisogna conoscere le coordinate X e Y di due vertici opposti del rettangolo da tracciare: per brevità, chiamiamo X1,Y1 le coordinate dell'angolo inferiore sinistro, e X2,Y2 quello dell'angolo superiore destro (il rettangolo può stare ovunque, sullo schermo, e le coordinate possono essere positive o negative). I lati del rettangolo sono tracciati con i comandi VAXY, VAX e VAY, che fanno muovere la tartaruga, dal punto in cui si trova, direttamente al punto specificato come dato di input del comando stesso, lasciando ogni altra cosa inalterata. Fra l'altro, l'orientamento della tartaruga non importa, e viene lasciato com'è. La procedura TRETTAN, dopo aver tracciato il rettangolo, riporta la tartaruga al punto di partenza, con il pennino sollevato.
Ecco le istruzioni:

```
PER TRETTAN :X1 :Y1 :X2 :Y2
ASSEGNA "X0 XCOR ASSEGNA "Y0 YCOR
SULAPENNA VAXY :X1 :Y1 GIULAPENNA
VAX :X2
VAY :Y2
VAX :X1
VAY :Y1
SULAPENNA VAXY :X0 :Y0
FINE
```

Le stesse coordinate X1,Y1 e X2,Y2 vengono usate per verificare se la tartaruga si trova dentro o fuori il rettangolo così definito. Se la tartaruga è all'interno del rettangolo, le sue coordinate X e Y sono comprese fra

X1 e X2, e fra Y1 e Y2, rispettivamente. La DENTRO? applica questo criterio, con un'unica istruzione:

```
PER DENTRO? :X1 :Y1 :X2 :Y2
RIPORTA (TUTTIVERI? (XCOR>:X1) (YCOR>:Y1) (XCOR<:X2)
   (YCOR<:Y2))
FINE
```

Il comando TUTTIVERI? vale "VERO se tutti i suoi dati di input valgono "VERO, altrimenti assume il valore "FALSO.
Se la tartaruga è all'esterno del rettangolo, la sua coordinata X è inferiore a X1 o superiore a X2, e/o la sua Y è inferiore a Y1, oppure superiore a Y2. La procedura FUORI? applica questo criterio, condensandolo anch'essa in una sola istruzione:

```
PER FUORI? :X1 :Y1 :X2 :Y2
RIPORTA (UNOVERO? (XCOR<:X1) (XCOR>:X2) (YCOR<:Y1)
   (YCOR>:Y2))
FINE
```

Il comando UNOVERO? passa in output il valore "VERO quando almeno uno dei suoi dati di input vale "VERO, mentre assume il valore "FALSO solo quando tutti i suoi dati di input valgono "FALSO.
Chi ha familiarità con gli operatori logici, riconoscerà che il comando UNOVERO? implementa l'operazione logica OR, e il comando TUTTIVERI? l'operazione logica AND.
Naturalmente, le procedure DENTRO? e FUORI? non hanno bisogno che il rettangolo da esaminare sia proprio disegnato sullo schermo, perché è sufficiente la sua definizione analitica, nel modo appena visto. Specialmente all'inizio, l'uso di queste procedure risulta più facile se ci si aiuta con schizzi su carta.

TRABOCCHETTO

TRETTAN, DENTRO? e FUORI? possono avere anche dati negativi, ma allora questi vanno posti fra parentesi, perché altrimenti il Logo cerca di eseguire l'operazione di sottrazione.

Ad esempio, il comando TRETTAN –10 –10 20 20 produce un messaggio di errore, in quanto il Logo esegue l'operazione –10 –10 = –20, e poi si trova con soli tre dati di input, invece dei quattro che occorrono.
Il comando TRETTAN (–10) (–10) 20 20, invece, funziona correttamente. Ancora un avvertimento: i primi due dati devono essere la coordinata X e la coordinata Y dell'angolo inferiore sinistro del rettangolo, e gli ultimi due la X e la Y dell'angolo superiore destro. Se non si fa così, le procedure si comportano in modo non corretto.

14.4 DISTANZA

La procedura DISTANZA fornisce in output il valore della distanza fra la tartaruga e un punto dato. Con essa, si può verificare anche se la tartaruga è all'interno o all'esterno di un dato cerchio, come si è visto nei giochi del tiro al bersaglio e della corsa in pista. La procedura si basa sul teorema di Pitagora, che si può esprimere così: "La lunghezza dell'ipotenusa di un triangolo rettangolo è pari alla radice quadrata della somma dei quadrati delle lunghezze dei cateti". In Figura 14.6, XCOR e YCOR sono le coordinate della tartaruga, mentre X1,Y1 sono quelle del punto che interessa.

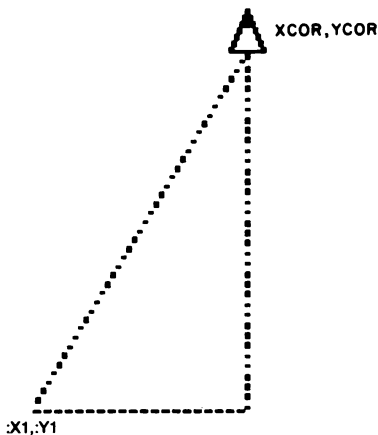



Figura 14.6 La distanza fra la tartaruga e il punto che interessa viene trovata in DISTANZA applicando il teorema di Pitagora

In senso orizzontale, la distanza fra il punto scelto e la tartaruga vale (XCOR–:X1), e in senso verticale (YCOR–:Y1). La distanza si trova calcolando la radice quadrata (RADQ) della somma dei quadrati di questi due valori. Anche la procedura DISTANZA è composta da una sola istruzione:

```
PER DISTANZA :X1 :Y1
RIPORTA RADQ ((XCOR-:X1)*(XCOR-:X1)+(YCOR-:Y1)*
    (YCOR-:Y1))
FINE
```

Secondo voi, sono necessarie le parentesi?

14.5 LEGGITASTO

La LEGGITASTO serve per leggere la battuta di un singolo tasto, senza interrompere la procedura che al momento è attiva: vengono usati due comandi del Logo, TASTO? e LEGGICAR.
LEGGICAR rimane in attesa che venga battuto un tasto, per passare in output il carattere corrispondente (proprio come fa LEGGIRIG, che però passa in output un'intera lista). Il comando TASTO? serve per rilevare che è stato premuto un tasto (in questo caso passa in output il valore "VERO, se no passa il valore "FALSO). Ecco le istruzioni di LEGGITASTO:

```
PER LEGGITASTO
SE TASTO? RIPORTA LEGGICAR
RIPORTA "
FINE
```

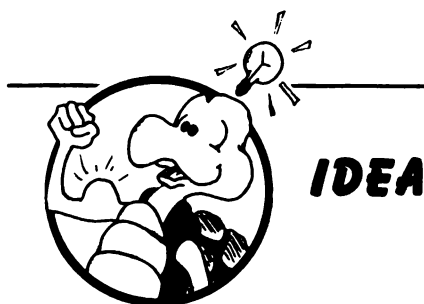
La traduzione in parole semplici di questa procedura è: "Se è stato battuto un tasto, passalo in output in forma di parola Logo; se no, passa in output una parola vuota".
La LEGGITASTO non fa nulla da sola: per funzionare, deve essere inserita in una procedura recursiva. Provatela pure da sola: non otterrete proprio nulla. Provatela invece inserita in una procedura come questa:

```
PER TASTAMPA
ASSEGNA "LETTERA LEGGITASTO
STAMPALETTERA :LETTERA
TASTAMPA
FINE
PER STAMPALETTERA :LET
SE :LET = "STOP
SE :LET = "A STAMPA [HAI BATTUTO A. ORA MI FERMO]
    PUNTOACAPO
STAMPA FRASE [HAI BATTUTO] :LET
FINE
```

La LEGGITASTO è di solito usata in una procedura di gestione dei comandi, in cui l'elaborazione prende strade diverse a seconda del tasto battuto.

14.6 DIMMI e QUALE

QUALE è una classica procedura Logo recursiva e dotata di output. Per molti, questo tipo di procedura è difficile da capire, ed è per questo che la descrizione compare solo alla fine del libro.

IDEA

Ecco due regole fondamentali per questo tipo di procedura:

1. La procedura deve terminare passando il suo output alla procedura o al comando che l'ha richiamata. Se a sua volta richiama un'altra procedura, rimane in attesa che questa termini, prima di proseguire con le istruzioni che ancora rimangono.
2. Una tecnica molto utile di programmazione di procedure recursive si basa sulla supposizione di conoscere già il modo di eseguire l'elaborazione con tutti i dati eccetto il primo. Dando poi per già pronta la parte che in realtà è ancora da fare, si incomincia con quella che ne sarà poi, in effetti, il completamento. Sembra strano? Lo è, ma funziona!

Consideriamo le due nuove procedure:

```
PER DIMMI :OGGETTO
ASSEGNA "NUMERO ACASO CONTA :OGGETTO
RIPORTA QUALE (:NUMERO+1) :OGGETTO
FINE
```

```
PER QUALE :NUMERO :OGGETTO
SE :NUMERO = 1 RIPORTA PRIMO :OGGETTO
RIPORTA QUALE (:NUMERO-1) (MENPRI :OGGETTO)
FINE
```

Vediamo, a parole, che cosa fanno le procedure: poi cercheremo di capire come.
QUALE ha due dati di input, un numero e un oggetto (che può essere una parola o una lista). Essa passa in output l'elemento *n*-esimo dell'oggetto (*n* è il primo dato di input).
DIMMI ha un solo dato di input, che può essere una parola o una lista. Essa passa in output un elemento a caso, dal suo input.
Consideriamo allora la QUALE:

```
PER QUALE :NUMERO :OGGETTO
SE :NUMERO = 1 RIPORTA PRIMO :OGGETTO
RIPORTA QUALE (:NUMERO-1) (MENPRI :OGGETTO)
FINE
```

Come si vede, è una procedura recursiva, che non si arresta fino a che non trova come primo dato il numero 1, oppure non prende il suo output da un'altra QUALE (il comando RIPORTA comprende anche uno STOP).
Seguiamo un esempio passo passo, distinguendo le singole procedure QUALE.
Pensiamo a questo caso:

```
QUALE 3 [A B C D]
```

Il Logo fornisce a QUALE due dati di input, il numero 3 e la lista [A B C D]. La QUALE riconosce che il suo primo dato di input non vale 1, perciò esegue la sua seconda istruzione, che dice di passare in output un valore ottenuto da una seconda procedura QUALE. Allora QUALE chiama QUALE2, dandole in input il numero 2 (inferiore di uno al proprio input) e la lista [B C D], cioè la lista che ha ricevuto meno il primo elemento. La QUALE rimane quindi in attesa di ricevere da QUALE2 il proprio output, da passare al Logo.
Gli input di QUALE2 sono 2 e [B C D]. Poiché il primo dato non è ancora il numero 1, essa esegue la seconda istruzione e richiama QUALE3, fornendo i dati di input 1 e [C D]. Poi rimane in attesa del valore da passare in output a QUALE.
La QUALE3 ha per input il numero 1 e la lista [C D]. Si limita quindi alla prima istruzione, e passa in output a QUALE2 il primo elemento della lista ricevuta in input, cioè la parola "C, poi si arresta.
La QUALE2, non appena riceve la parola "C, termina la sua seconda

istruzione passando "C alla QUALE, e poi si arresta.
La QUALE, a sua volta, appena riceve la "C da QUALE2, termina la sua seconda istruzione passando la stessa "C a Logo, e poi si arresta.
Di solito, la QUALE è usata per fornire direttamente l'input a un'altra procedura, o a un comando:

```
STAMPA QUALE 3 [A B C D]
C
```

In questo esempio, come si può vedere, C è il terzo elemento della lista di input di QUALE. Se il primo dato di QUALE fosse 2, oppure 4, si otterrebbe in output B, oppure D, rispettivamente. Se il primo dato di QUALE fosse maggiore del numero di elementi della lista, il Logo protesterebbe, perché si arriverebbe ad avere una lista vuota, nel processo recursivo di QUALE.
La DIMMI impiega il comando ACASO del Logo, e la procedura QUALE, ma non è recursiva:

```
PER DIMMI :OGGETTO
ASSEGNA "NUMERO ACASO CONTA :OGGETTO
RIPORTA QUALE (:NUMERO+1) :OGGETTO
FINE
```

Vediamo un esempio:

```
DIMMI [A B C D]
```

Nella prima istruzione, CONTA [A B C D] fornisce il numero 4, che viene dato in input ACASO. Con ACASO 4, si estrae un numero a caso, compreso fra 0 e 3 (supponiamo di trovare 2).
Nella seconda istruzione, si trova QUALE 3 [A B C D], cioè la lettera C, che viene subito passata in output.
Ecco cosa succede:

```
STAMPA DIMMI [A B C D]
C
```

Se ci riproviamo, abbiamo tre probabilità su quattro di vedere un risultato diverso. Si può provare anche molte volte, ad esempio con questo comando:

```
RIPETI 10[STAMPA DIMMI [A B C D]]
```

14.7 LEGGINUMERO

Con LEGGINUMERO, troviamo un altro caso di procedura recursiva con output. Essa ripara una piccola mancanza del Logo, che dispone del solo comando LEGGIRIG, per leggere da tastiera più di un carattere. L'output di LEGGIRIG ha però forma di lista, che non va bene per rappresentare numeri, che sono invece un tipo speciale di parola. Dato che lista e parola sono eterogenee fra loro, bisogna costruire una procedura che legga da tastiera un numero, interpretandolo come tale, cioè come parola. Provate questo

```
ASSEGNA "NUMERO LEGGIRIG
55
STAMPA :NUMERO
55
```

Sembra tutto a posto. Però:

```
STAMPA (:NUMERO = 55)
FALSO
```

Ma certo! In realtà :NUMERO è la lista [55], che in Logo non corrisponde affatto al numero 55. Per ottenere un numero da [55], dobbiamo estrarre il primo elemento di tale lista:

```
STAMPA (PRIMO :NUMERO) = 55
VERO
```

Il compito principale di LEGGINUMERO consiste quindi nell'estrarre il primo elemento dell'output di LEGGIRIG:

```
PER LEGGINUMERO
RIPORTA PRIMO LEGGIRIG
FINE
```

Saremmo a posto così, se fossimo sicuri che possono essere battuti solo numeri alla tastiera. Però così non è, perché chi sta alla tastiera si può sbagliare, battendo magari solo il tasto RETURN. La LEGGIRIG fornirebbe allora in output una lista vuota [ ], e LEGGINUMERO darebbe un messaggio di errore:

```
PRIMO NON ACCETTA [ ] COME ARGOMENTO
```

Che fare per evitare tutto ciò e fare in modo che venga battuto proprio un numero? Si può modificare LEGGINUMERO, usando il comando NUMERO? del Logo:

```
PER LEGGINUMERO
ASSEGNA "NUM1 LEGGIRIG
VERIFICA :NUM1 = [ ]
SEVERO STAMPA [CI VUOLE UN NUMERO] RIPORTA
   LEGGINUMERO
VERIFICA NON NUMERO? PRIMO :NUM1
SEVERO STAMPA [CI VUOLE UN NUMERO] RIPORTA
   LEGGINUMERO
SEFALSO RIPORTA PRIMO :NUM1
FINE
```

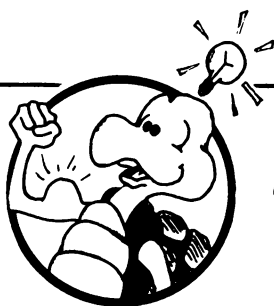
Vediamo la nuova versione passo passo, ricordando che la prima regola per procedure con output è che ogni chiamata deve concludersi con un output.
La prima istruzione assegna il nome "NUM1 alla lista di output di LEGGIRIG (che passa, naturalmente, quello che viene battuto alla tastiera, così com'è).
LEGGINUMERO esegue quindi due verifiche, per vedere se :NUM1 è una lista vuota, oppure se il suo primo elemento non è un numero. Se entrambe le verifiche hanno successo (devono dare "FALSO) viene eseguita l'ultima istruzione, che passa in output PRIMO :NUM1 (questa è la strada seguita più spesso).
Se invece :NUM1 è una lista vuota, oppure PRIMO :NUM1 non è un numero, uno dei due test vale "VERO, e allora il computer stampa "CI VUOLE UN NUMERO" e richiama una nuova LEGGINUMERO, rimanendo in attesa che questa termini per passare in output l'output stesso di questa.
Questo processo va avanti fino a quando non viene battuto alla tastiera un numero (si può anche arrestare quando viene occupata tutta la memoria disponibile del computer). Quando viene finalmente battuto un numero, questo passa in output alla procedura che lo ha richiesto, e da questa all'output di quella che l'ha richiamata, e così via fino alla procedura che ha richiamato la prima LEGGINUMERO (proprio come fa la QUALE, vista nel paragrafo 14.6).
Provate a battere più volte il tasto RETURN:

```
STAMPA LEGGINUMERO
CI VUOLE UN NUMERO
CI VUOLE UN NUMERO
```

CI VUOLE UN NUMERO

Il messagio viene ripetuto fino a quando non si batte un numero, oppure non si ferma tutto battendo CTRL-G. Si ottiene lo stesso risultato anche se si battono tasti che non corrispondono a numeri:

```
STAMPA LEGGINUMERO
SETTE
CI VUOLE UN NUMERO
7
7
```

In un certo senso, LEGGINUMERO è più semplice di QUALE o CONTA, perché è recursiva solo in caso di errore, quando non viene battuto un numero alla tastiera.

IDEA

LEGGINUMERO è un esempio di procedura a prova di errore, che cioè riesce a proteggere il programma in cui è inserita, da alcuni errori previsti in anticipo. Se si usa infatti la prima versione di LEGGINUMERO, si corre il rischio di interrompere il programma che la usa, ogni volta che per sbaglio non si batte un numero.
I giochi di animazione e i quiz dovrebbero essere tutti realizzati con questo criterio, in modo da non interromperli troppo spesso per un errore banale di battitura. Se siete interessati all'argomento, potete leggere il libro di Scot Kamins e Mitchell Waite, *Programmazione umanizzata in Applesoft* pubblicato dalla McGraw-Hill (anche se il libro è orientato al linguaggio BASIC, molte considerazioni su come adattare un programma all'uomo valgono in generale, e quindi anche per il Logo).

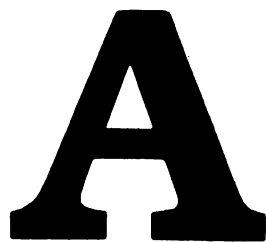
14.8 ASPETTA

La più semplice delle procedure di servizio è ASPETTA, che distrae il computer per un po':

```
PER ASPETTA :TEMPO
SE :TEMPO = 0 STOP
ASPETTA :TEMPO-1
FINE
```

Come si vede, ASPETTA verifica se :TEMPO vale 0, per arrestarsi; in caso contrario richiama un'altra ASPETTA riducendo di uno il dato di input che assegna a questa. In pratica, l'esecuzione di queste semplici istruzioni prende un po' di tempo, che diventa apprezzabile se l'input di ASPETTA è grande. Con ASPETTA 100, il computer è occupato per circa due secondi, prima di passare al comando successivo.
Per capire meglio come funziona la procedura, provate questa:

```
PER CONTOALLAROVESCIA :TEMPO
STAMPA :TEMPO
SE :TEMPO = 0 STOP
CONTOALLAROVESCIA :TEMPO-1
FINE
```

Con CONTOALLAROVESCIA 100 si stampano i numeri da 100 a zero, e poi la procedura si arresta. Con ASPETTA 100 si fa la stessa cosa, senza però stampare nulla. (La CONTOALLAROVESCIA prende un po' più di tempo, rispetto ad ASPETTA, per via dell'operazione di scrittura sullo schermo).

Come preparare il disco di procedure

Viene qui descritto come preparare un proprio disco di procedure. Per riuscire in questo scopo, occorre però avere già familiarità con l'editor e con gli strumenti di gestione dei file del Logo (non importa in quale versione). In altre parole, bisogna avere padronanza dell'argomento del Capitolo 4 di questo libro. È utile anche la conoscenza della sintassi dei comandi Logo, specie per quanto riguarda l'impiego delle variabili (questo argomento è trattato nei Capitoli 7 e 9).
Il disco comprende quattordici file, contenenti ciascuno le procedure (da una sola a venti) che svolgono una certa funzione, e che vanno tenute separate da quelle degli altri file. Occorre perciò ripulire sempre la memoria del computer prima di iniziare la preparazione di un nuovo file.
Tutte le procedure presentate qui sono state provate a lungo, con l'intento di eliminare tutti gli errori. È possibile però che ricopiandole si commetta qualche errore di battitura: ogni procedura va perciò provata di nuovo, prima di registrarla nella versione finale del disco. Per ogni procedura, viene anche indicata la parte del libro in cui questa viene usata, in modo da facilitare il debugging.

A.1 CERCHI

Ripulite la memoria del computer, battendo CANCELLA TUTTO oppure CIAO.
Ricopiate poi queste istruzioni:

```
PER CERCHIODES :R
RIPETI 36 [RCP :R]
FINE
PER ARCODES :R
RIPETI 9 [RCP :R]
FINE
PER RCP :R
DESTRA 5
AVANTI :R*(3.14159)/18
DESTRA 5
FINE
PER CERCHIOSIN :R
RIPETI 36 [LCP :R]
FINE
PER ARCOSIN :R
RIPETI 9 [LCP :R]
FINE
PER LCP :R
SINISTRA 5
AVANTI :R*(3.14159)/18
SINISTRA 5
FINE
```

Al termine della battitura, uscite dall'editor battendo CTRL-C, ritornando così all'ambiente normale del Logo. Provate le procedure, con gli esempi del Capitolo 2. Se tutto va bene, registratele sul vostro disco con il comando:

```
CONSERVA "CERCHI
```

A.2 CERCHIETTO

Ripulite la memoria del computer, battendo CANCELLA TUTTO oppure CIAO.
Ricopiate poi queste istruzioni:

```
PER CERCHIETTO :R
ASSEGNA "PEN? PRIMO STATOSCHERMO
ASSEGNA "VISTA? PRIMO MENPRI STATOSCHERMO
NASTARTA
SULAPENNA AVANTI :R
DESTRA 90
```

```
GIULAPENNA RIPETI 36 [RCP :R]
SINISTRA 90
SULAPENNA INDIETRO :R
SE :PEN? GIULAPENNA
SE :VISTA? MOSTARTA
FINE
PER RCP :R
DESTRA 5
AVANTI :R*3.14159/18
DESTRA 5
FINE
```

Alla fine, battete CTRL-C per uscire dall'editor e ritornare nell'ambiente normale del Logo. Provate le procedure con gli esempi del Capitolo 10 e, quando tutto funziona per bene, registratele su disco battendo:

```
CONSERVA "CERCHIETTO
```

A.3 RETTANGOLI

Ripulite la memoria del computer con il comando CANCELLA TUTTO oppure CIAO.
Ricopiate poi queste istruzioni:

```
PER TRETTAN :X1 :Y1 :X2 :Y2
ASSEGNA "X0 XCOR ASSEGNA "Y0 YCOR
SULAPENNA VAXY :X1 :Y1 GIULAPENNA
VAX :X2
VAY :Y2
VAX :X1
VAY :Y1
SULAPENNA VAXY :X0 :Y0
FINE
```

Le due procedure DENTRO? e FUORI? comprendono entrambe un'istruzione molto lunga, che va battuta senza interromperla con il tasto RETURN, che va battuto solo alla fine dell'istruzione:

```
PER FUORI? :X1 :Y1 :X2 :Y2                              (tasto RETURN)
RIPORTA (UNOVERO? (XCOR<:X1) (XCOR>:X2)
    (YCOR<:Y1) (YCOR>:Y2))                              (tasto RETURN)
FINE                                                    (tasto RETURN)
```

```
PER DENTRO? :X1 :Y1 :X2 :Y2                          (tasto RETURN)
RIPORTA (TUTTIVERI? (XCOR>:X1) (YCOR>:Y1)
    (XCOR<:X2) (YCOR<:Y2))                           (tasto RETURN)
FINE                                                 (tasto RETURN)
```

Quando avete finito, abbandonate l'editor battendo CTRL-C. Ritornati così nell'ambiente normale del Logo, provate le procedure, con gli esempi del Capitolo 12. Quando tutto funziona bene, registrate le procedure su disco, con:

```
CONSERVA "RETTANGOLI
```

A.4 DISTANZA

Ripulite la memoria del computer, con CANCELLA TUTTO oppure con CIAO.
Ricopiate le istruzioni (battendo il tasto RETURN solo quando è indicato):

```
PER DISTANZA :X1 :Y1                                 (tasto RETURN)
RIPORTA RADQ ((XCOR-:X1)*(XCOR-:X1)+
    (YCOR-:Y1)*(YCOR-:Y1))                           (tasto RETURN)
FINE                                                 (tasto RETURN)
```

Al termine, battete CTRL-C. per abbandonare l'editor e ritornare nel normale ambiente Logo. Provate poi le procedure secondo gli esempi del Capitolo 10. Quando funziona tutto bene, registratele su disco battendo:

```
CONSERVA "DISTANZA
```

A.5 LEGGITASTO

Ripulite la memoria del computer, con CANCELLA TUTTO o con CIAO.
Ricopiate poi queste istruzioni:

```
PER LEGGITASTO
SE TASTO? RIPORTA LEGGICAR
RIPORTA "
FINE
```

Provate la procedura (dopo aver abbandonato l'editor con CTRL-C, seguen-

do gli esempi del paragrafo 14.5. Registratela su disco quando siete sicuri che funziona bene, con:

```
CONSERVA "LEGGITASTO
```

A.6 DIMMI

Ripulite la memoria del computer con il comando CANCELLA TUTTO oppure con CIAO.
Ricopiate poi queste istruzioni:

```
PER DIMMI :OGGETTO
ASSEGNA "NUMERO ACASO CONTA :OGGETTO
RIPORTA QUALE (:NUMERO+1) :OGGETTO
FINE
PER QUALE :NUMERO :OGGETTO
SE :NUMERO = 1 RIPORTA PRIMO :OGGETTO
RIPORTA QUALE (:NUMERO-1) (MENPRI :OGGETTO)
FINE
```

Abbandonate poi l'editor (battendo CTRL-C) e provate le procedure, seguendo gli esempi del Capitolo 13. Quando tutto funziona bene, registratele su disco con:

```
CONSERVA "DIMMI
```

A.7 LEGGINUMERO

Ripulite la memoria del computer, battendo CANCELLA TUTTO o anche CIAO.
Ricopiate poi queste istruzioni:

```
PER LEGGINUMERO
ASSEGNA "NUM1 LEGGIRIG
VERIFICA :NUM1 = [ ]
SEVERO STAMPA [CI VUOLE UN NUMERO] RIPORTA
    LEGGINUMERO
VERIFICA NON NUMERO? PRIMO :NUM1
SEVERO STAMPA [CI VUOLE UN NUMERO] RIPORTA
    LEGGINUMERO
```

```
SEFALSO RIPORTA PRIMO :NUM1
FINE
```

Uscite poi dall'ambiente dell'editor (battendo CTRL-C), per ritornare al normale ambiente Logo. Provate la procedura per bene, secondo gli esempi del Capitolo 9, per registrarla poi su disco con:

```
CONSERVA "LEGGINUMERO
```

A.8 ASPETTA

Ripulite la memoria del computer con CANCELLA TUTTO o con CIAO. Ricopiate poi queste istruzioni:

```
PER ASPETTA :TEMPO
SE :TEMPO = 0 STOP
ASPETTA :TEMPO-1
FINE
```

Provate la procedura, secondo le indicazioni del Capitolo 10 (per uscire dall'editor, si batte CTRL-C) e registratela su disco con il comando:

```
CONSERVA "ASPETTA
```

A.9 INDOVINUMERO

Ripulite la memoria del computer, battendo CANCELLA TUTTO oppure CIAO.
Ricuperate poi da disco la procedura LEGGINUMERO, battendo:

```
RECUPERA "LEGGINUMERO
```

Ricopiate quindi queste istruzioni:

```
PER INDOVINUMERO
PRESENTO
ESTRAGGO
PRENDO
FINE
```

```
PER PRESENTO
PULISCITESTO
STAMPA [STO PENSANDO A UN NUMERO FRA 0 E 100!]
STAMPA [INDOVINA QUAL E'?]
FINE
PER ESTRAGGO
ASSEGNA "NUMERO 1+ACASO 99
FINE
PER PRENDO
SCRIVI ">
ASSEGNA "TENTATIVO LEGGINUMERO
CONFRONTO :TENTATIVO :NUMERO
FINE
PER CONFRONTO :TENTATIVO :NUMERO
SE :TENTATIVO = :NUMERO STAMPA [INDOVINATO!] STOP
SE :TENTATIVO > :NUMERO STAMPA [NO: CALA] PRENDO STOP
SE :TENTATIVO < :NUMERO STAMPA [NO: AUMENTA] PRENDO
   STOP
FINE
```

Al termine, battete CTRL-C per abbandonare l'editor, e provate le procedure seguendo le indicazioni del Capitolo 9. Poi registratele su disco con:

```
CONSERVA "INDOVINUMERO
```

A.10 MATQUIZ

Ripulite la memoria del computer con il comando CANCELLA TUTTO oppure con CIAO.
Ricuperate poi da disco la procedura LEGGINUMERO, battendo:

```
RECUPERA "LEGGINUMERO
```

Ricopiate quindi queste istruzioni:

```
PER MATQUIZ
PULISCITESTO
QUANTEDOMANDE
ASSEGNA "CONTO 1
ASSEGNA "PUNTI 0
NUOVOQUIZ :CONTO :QUANTE :PUNTI
FINE
```

```
PER QUANTEDOMANDE
STAMPA [QUANTE DOMANDE VUOI PROVARE?]
ASSEGNA "QUANTE LEGGINUMERO
FINE
PER NUOVOQUIZ :CONTO :QUANTE :PUNTI
PULISCITESTO
ADDENDI
DOMANDO
CHESOMMA
ASPETTO
SE :CONTO = :QUANTE MIFERMO STOP
NUOVOQUIZ (:CONTO+1) :QUANTE :PUNTI
FINE
PER ADDENDI
ASSEGNA "A1 ACASO 100
ASSEGNA "A2 ACASO 100
ASSEGNA "SOMGIUSTA :A1+:A2
FINE
PER DOMANDO
STAMPA FRASE [DOMANDA] :CONTO
STAMPA [ ]
SCRIVI (FRASE :A1 [+] :A2 [=])
FINE
PER CHESOMMA
ASSEGNA "SOMMA LEGGINUMERO
VERIFICA :SOMMA = :SOMGIUSTA
SEVERO STAMPA [GIUSTO]
SEVERO ASSEGNA "PUNTI :PUNTI+1
SEFALSO STAMPA FRASE [NO: LA RISPOSTA E'] :SOMGIUSTA
FINE
PER ASPETTO
STAMPA [PER FAVORE, BATTI IL TASTO RETURN]
STAMPA LEGGIRIG
FINE
PER MIFERMO
PULISCITESTO
STAMPA FRASE [PUNTEGGIO OTTENUTO:] :PUNTI
STAMPA (FRASE [SU] :QUANTE [DOMANDE])
FINE
```

Al termine, battete CTRL-C per abbandonare l'editor e provate le procedure, seguendo le indicazioni del Capitolo 9. Registratele su disco con il comando:

```
CONSERVA "MATQUIZ
```

A.11 TIRO.A.SEGNO

Ripulite la memoria del computer, battendo CANCELLA TUTTO oppure CIAO.
Caricate poi le procedure di servizio CERCHIETTO, DISTANZA, LEGGINUMERO e ASPETTA, che dovrebbero già trovarsi sul vostro disco; se non ci sono, ricopiatele e registratele separatamente sul disco, prima di passare a TIRO.A.SEGNO. Le procedure si leggono con i comandi:

```
RECUPERA "CERCHIETTO
RECUPERA "DISTANZA
RECUPERA "LEGGINUMERO
RECUPERA "ASPETTA
```

Ricopiàte poi le seguenti istruzioni:

```
PER VIA
INIZIADATI
INCOMINCIA
FINE
PER INIZIADATI
SEME
ASSEGNA "COLPI 0
ASSEGNA "XBERS (90-10*ACASO 19)
ASSEGNA "YBERS (80-10*ACASO 6)
ASSEGNA "XPART (90-10*ACASO 19)
ASSEGNA "YPART (-10*ACASO 3)
ASSEGNA "OPART (10*ACASO 36)
FINE
PER INCOMINCIA
DISEGNA
NASTARTA
TRACCIABERSAGLIO :XBERS :YBERS
TARTARUGAVAI :XPART :YPART :OPART
MOSTARTA
FINE
PER TRACCIABERSAGLIO :XBERS :YBERS
SULAPENNA
VAXY :XBERS :YBERS
CERCHIETTO 10
FINE
PER TARTARUGAVAI :XPART :YPART :OPART
SULAPENNA
```

```
VAXY :XPART :YPART
ASDIR :OPART
FINE
PER TIRO
ASSEGNA "COLPI :COLPI+1
STAMPA [A CHE DISTANZA?]
ASSEGNA "GITTATA LEGGINUMERO
GIULAPENNA AVANTI :GITTATA
VERIFICA DISTANZA :XBERS :YBERS<10
SEVERO COLPITO
SEFALSO MANCATO
FINE
PER COLPITO
STAMPA [CONGRATULAZIONI! BERSAGLIO COLPITO!]
STAMPA (FRASE [CI SONO VOLUTI SOLO] :COLPI [COLPI])
FINE
PER MANCATO
STAMPA FRASE [MANCATO! COLPI TIRATI:] :COLPI
ASPETTA 200
TARTARUGAVAI :XPART :YPART :OPART
FINE
```

Battete CTRL-C, per lasciare l'editor e provate le procedure, come nel Capitolo 3. Registratele poi con:

```
CONSERVA "TIRO.A.SEGNO
```

A.12 SCHIZZO

Ripulite la memoria del computer, con CANCELLA TUTTO oppure CIAO. Caricate poi in memoria la procedura di servizio LEGGITASTO, che dovrebbe essere ormai pronta sul vostro disco (se non lo è, registratela prima di passare a SCHIZZO). Il recupero di LEGGITASTO da disco si fa con:

```
RECUPERA "LEGGITASTO
```

Ricopiate poi le istruzioni seguenti:

```
PER M
VIA
SCHIZZO
FINE
PER VIA
ASSEGNA "LISCOM [ ]
DISEGNA
FINE
PER SCHIZZO
COMANDO
SCHIZZO
FINE
PER COMANDO
ASSEGNA "COMAN LEGGITASTO
SE :COMAN = "A AVANTI 20 AGGLETTERA :COMAN
SE :COMAN = "I INDIETRO 20 AGGLETTERA :COMAN
SE :COMAN = "D DESTRA 30 AGGLETTERA :COMAN
SE :COMAN = "S SINISTRA 30 AGGLETTERA :COMAN
SE :COMAN = "Z BASTA PUNTOACAPO
FINE
PER AGGLETTERA :LET
ASSEGNA "LISCOM FRASE :LISCOM :LET
FINE
PER BASTA
DISCHERMO
STAMPA [COME CHIAMI IL DISEGNO?]
STAMPA [SE LO VUOI BUTTARE,]
STAMPA [BATTI SOLO IL TASTO RETURN]
ASSEGNA "RISPOS LEGGIRIG
SE :RISPOS = [ ] STOP
ASSEGNA (PRIMO :RISPOS) :LISCOM
FINE
PER RD :LISCOM
SE :LISCOM = [ ] STOP
RICOMANDO PRIMO :LISCOM
RD MENPRI :LISCOM
FINE
PER RICOMANDO :COMAN
SE :COMAN = "A AVANTI 20
SE :COMAN = "I INDIETRO 20
SE :COMAN = "D DESTRA 30
SE :COMAN = "S SINISTRA 30
FINE
```

Battete poi CTRL-C,per lasciare l'editor e provate le procedure, come è descritto nel Capitolo 3. Registratele quindi su disco con il comando:

```
CONSERVA "SCHIZZO
```

A.13 CORSA

Ripulite la memoria di lavoro con CANCELLA TUTTO oppure con CIAO. Caricate poi dal disco le procedure di servizio CERCHIETTO, DISTANZA e LEGGITASTO (se queste non sono già state registate, fatelo, prima di passare a CORSA). Il caricamento in memoria da disco si fa con i comandi:

```
RECUPERA "CERCHIETTO
RECUPERA "DISTANZA
RECUPERA "LEGGITASTO
```

Ricopiate poi le seguenti istruzioni:

```
PER CORSA
PISTA
PARTENZA
CORRI 0
FINE
PER PISTA
DISEGNA
NASTARTA
CERCHIETTO 50
CERCHIETTO 70
SINISTRA 90
SULAPENNA AVANTI 50
GIULAPENNA AVANTI 20
SULAPENNA INDIETRO 70
DESTRA 90
FINE
PER PARTENZA
VAXY (-60) 0
ASDIR 0
AVANTI 1
MOSTARTA
ASSEGNA "YVEC 1
ASSEGNA "META 0
FINE
```

```
PER CORRI :TEMPO
SE FUORIPISTA? AUTOFUORI STOP
SE FINECORSA? ARRIVO STOP
AVANTI :META
COMANDO
CORRI :TEMPO+1
FINE
PER FUORIPISTA?
SE (DISTANZA 0 0)>70 RIPORTA "VERO
SE (DISTANZA 0 0)<50 RIPORTA "VERO
RIPORTA "FALSO
FINE
PER AUTOFUORI
STAMPA [SEI USCITO DI PISTA!]
FINE
PER FINECORSA?
SE TUTTIVERI? (YCOR>0) (:YVEC<0) RIPORTA "VERO
ASSEGNA "YVEC YCOR
RIPORTA "FALSO
FINE
PER ARRIVO
STAMPA [SEI ARRIVATO AL TRAGUARDO]
STAMPA FRASE [CON IL TEMPO DI] :TEMPO
FINE
PER COMANDO
ASSEGNA "COMAN LEGGITASTO
SE :COMAN = " STOP
SE :COMAN = "V ASSEGNA "META :META+5 STOP
SE :COMAN = "L ASSEGNA "META :META-5 STOP
SE :COMAN = "D DESTRA 30 STOP
SE :COMAN = "S SINISTRA 30 STOP
FINE
PER RIPARTI
PARTENZA
CORRI 0
FINE
```

Battete CTRL-C, per lasciare l'editor e provate attentamente le procedure, per registrarle poi su disco con:

```
CONSERVA "CORSA
```

A.14 POETA

Ripulite la memoria del computer battendo CANCELLA TUTTO oppure CIAO.
Caricate poi in memoria le due procedure di servizio LEGGINUMERO e DIMMI, che dovrebbero essere già registrate sul vostro disco: se non lo sono, fatelo prima di passare a POETA. Il caricamento in memoria delle procedure si fa battendo:

```
RECUPERA "DIMMI
RECUPERA "LEGGINUMERO
```

Battete poi le istruzioni riportate qui di seguito (attenzione, che le variabili "ARTICOLI, "AGGETTIVI, "NOMI, "VERBI, "PREPOSIZIONI e "AVVERBI non indicano procedure, ma liste, da creare con il comando ASSEGNA prima di lasciare l'editor):

```
PER POESIE
PULISCITESTO
STAMPA [QUANTE POESIE VUOI?]
ASSEGNA "N LEGGINUMERO
PULISCITESTO STAMPA [ ] STAMPA [ ]
STAMPA FRASE :N [LOGOPOESIE]
STAMPA [ ]
RIPETI :N [POETA STAMPA [ ]]
FINE
PER POETA
STAMPA VERSO1
STAMPA VERSO2
STAMPA VERSO3
FINE
PER VERSO1
RIPORTA (FRASE ARTICOLO AGGETTIVO NOME)
FINE
PER VERSO2
RIPORTA (FRASE ARTICOLO NOME VERBO PREPOSIZIONE
   ARTICOLO NOME AGGETTIVO)
FINE
PER VERSO3
RIPORTA (FRASE AGGETTIVO AGGETTIVO NOME)
FINE
PER ARTICOLO
RIPORTA DIMMI :ARTICOLI
```

```
FINE
PER NOME
RIPORTA DIMMI :NOMI
FINE
PER AGGETTIVO
RIPORTA DIMMI :AGGETTIVI
FINE
PER VERBO
RIPORTA DIMMI: VERBI
FINE
PER PREPOSIZIONE
RIPORTA DIMMI :PREPOSIZIONI
FINE
PER DIZIONARIO
PULISCITESTO
STAMPA [ARTICOLI:]
STAMPA [ ] STAMPA :ARTICOLI
STAMPA [ ]
STAMPA [NOMI:]
STAMPA [ ] STAMPA :NOMI
STAMPA [ ]
STAMPA [VERBI:]
STAMPA [ ] STAMPA :VERBI
STAMPA [ ]
STAMPA [PER CONTINUARE, BATTI IL TASTO RETURN]
STAMPA LEGGIRIG
PULISCITESTO
STAMPA [AGGETTIVI:]
STAMPA [ ] STAMPA :AGGETTIVI
STAMPA [ ]
STAMPA [PREPOSIZIONI:]
STAMPA [ ] STAMPA :PREPOSIZIONI
STAMPA [ ]
STAMPA [AVVERBI:]
STAMPA [ ] STAMPA :AVVERBI
FINE
ASSEGNA "ARTICOLI [IL LO LA UN UNO UNA]
ASSEGNA "AGGETTIVI [AUTUNNALE NASCOSTO FRIZZANTE
    RIBOLLENTE VORTICOSO VERDE AMARO NEBBIOSO
    SILENZIOSO VUOTO LIMPIDO SELVAGGIA NERO BIANCO
    VIOLA BLU CALMO IMMOBILE FORTE DEBOLE ANTICO
    NUOVO ALTO]
ASSEGNA "NOMI [CASCATA FIUME BREZZA LUNA PIOGGIA VENTO
    MARE MATTINO NEVE LAGO TRAMONTO OMBRA PINO
```

```
        FOGLIA LUCCICHIO ALBA BOSCO STAGNO RONDINE ABETE
        POLVERE NOTTE CORVO CIVETTA SERA COLLE BREZZA
        ERBA MONTE VALLE FIOCCO VOLPE LUPO ORSO GABBIANO
        DESIDERIO MEMORIA]
    ASSEGNA "VERBI [SCUOTE PORTA [SI NASCONDE] DORME [SI
        ALLUNGA] MORMORA VOLA VIBRA CADE SFRECCIA SALTA
        PIANGE RIDE PREGA FIORISCE SENTE SCOPRE CIGOLA
        STRIDE RISUONA ECHEGGIA TACE RICORDA TREMA]
    ASSEGNA "PREPOSIZIONI [SOTTO SOPRA FUORI DENTRO OLTRE
        PRESSO CON PER FRA DI TRA [VICINO A] [INTORNO A] IN DA]
    ASSEGNA "AVVERBI [RAPIDAMENTE LENTAMENTE
        ALLEGRAMENTE TRISTEMENTE QUIETAMENTE
    DOLCEMENTE INVANO]
```

Battete infine CTRL-C per lasciare l'editor. Provate le procedure per bene, e registratele su disco con:

```
    CONSERVA "POETA
```

Ora avete anche voi il vostro disco di procedure.

B

Come trattare i dischi e i file

Viene qui descritto il modo di inizializzare, conservare e copiare i dischi di servizio del Logo (con alcuni consigli su come vanno trattati i dischi). Bisogna sempre tenere presente che la cosa più importante del vostro Logo è l'insieme delle procedure e dei dati che voi, man mano, andate creando. Questa massa di informazioni è registrata su disco, ed è esposta ai danni che derivano dal caso, o anche dalla vostra incuria. Il contenuto dei dischi è sempre modificabile, o cancellabile: perciò è importante capire come lo si può proteggere e conservare, tenendo magari anche una seconda copia dei dischi più importanti.

B.1 Come inizializzare un disco di servizio per il Logo

I dischi di servizio per il Logo possono essere inizializzati con i comandi BASIC, come qualunque altro disco per Commodore 64. Per inizializzare un nuovo disco, si seguono quindi i seguenti passi:

1. Accendere il drive, il monitor o la TV e il computer (in questo ordine).
2. Far partire il Logo, e poi estrarre il disco Logo dal drive.
3. Inserire nel drive il disco che si vuole inizializzare (che può essere un disco nuovo, o anche un disco vecchio, il cui contenuto non interessa più). Quando un disco è inizializzato, viene cancellata ogni informazione in esso contenuta.

4. Battere:

   DOS [N0:NOMEIO,64] (e RETURN)

   Al posto di NOMEIO va bene qualunque altra frase, lunga al massimo 16 caratteri. Al posto di 64, va bene qualunque altro numero di due cifre.
5. Quando compare sullo schermo il simbolo ? si estrae dal drive il disco, ormai inizializzato. Conviene scrivere il nome sull'etichetta (con una penna a feltro: non scrivere mai con matite o penne biro, perché si può danneggiare il disco).

B.2 Come copiare un disco di servizio del Logo

A volte è utile copiare un disco intero, per ottenere ad esempio una copia di riserva (non si sa mai...), oppure per dare a un amico le proprie procedure da provare. La copia di un disco si può fare con uno o con due drive, procedendo però in modo diverso nei due casi.

COPIA CON UN SOLO DRIVE

Per copiare un disco usando un solo drive, occorre il programma "1541 Backup", disponibile nel disco di utility per il C-64, Disk Bonus Pack C64103.
Con il programma a disposizione, si eseguono le seguenti manovre:

1. Accendere il drive e poi il computer, come al solito, battendo quindi:

   LOAD"1541 BACKUP",8 (e RETURN)

   Il computer scrive allora:

   SEARCHING FOR 1541 BACKUP
   LOADING
   READY

2. Quando si vede la parola READY e il cursore si mette a lampeggiare, battere:

   RUN

3. Lo schermo diventa bianco, e presenta sei riquadri: l'ultimo, che porta la scritta Operator Intervention, indica le manovre da eseguire.

Quando il programma si avvia, compare un cursore lampeggiante nel riquadro 1, mentre nel riquadro 6 viene visualizzata (in blu) la scritta:

ENTER THE PROGRAM OPERATION CODE

Bisogna battere il tasto RETURN.

4. Il cursore si sposta ora nel riquadro 3, dove è scritto DESTINATION. Nel riquadro 6 compare la scritta:

   ENTER DISK NAME

   Bisogna battere il nome da assegnare al nuovo disco (lungo fino a 16 caratteri), terminando con il tasto RETURN.
5. Il cursore va ora in fondo al riquadro 3 e la scritta del riquadro 6 diventa:

   ENTER DISK ID NUMBER

   Bisogna battere un numero di due cifre (un qualunque numero da 01 a 99 va bene), e poi il tasto RETURN.
6. Compare ora (in rosso) questa scritta nel riquadro 6:

   INSERT DESTINATION DISK

   Si estrae dal drive il disco con il programma di copia e si inserisce il disco nuovo, battendo poi il tasto RETURN.
7. Ora compare nel riquadro 5 (intitolato Executing) la scritta:

   FORMATTING DESTINATION DISK

   L'operazione dura un paio di minuti.
8. Al termine della formattazione del nuovo disco, compare nell'ultimo riquadro la scritta (in rosso):

   INSERT SOURCE DISK INTO DRIVE

   Si estrae dal drive il disco nuovo, si inserisce il disco originale (quello che va duplicato) e si batte il tasto RETURN. Nel riquadro 5 compare la scritta (in verde):

   READING BAM FROM SOURCE DISK

9. Compare poi nel riquadro 6 la scritta:

VERIFY SOURCE DISK FOR BACKUP

Se il disco nel drive è proprio quello da duplicare, si batte il tasto RETURN.

10. Nel riquadro 5 compare ora la scritta (in verde):

    READING DATA INTO BUFFER

    e nel riquadro 2 si vede l'immagine del buffer che si sta riempiendo (in questa fase, il contenuto del disco viene trasferito nella memoria del computer).

11. Quando il buffer è pieno, nel riquadro 6 compare la scritta:

    INSERT DESTINATION DISK

    Si estrae il disco originale dal drive, si inserisce il disco nuovo e si batte il tasto RETURN. Si vede allora nel riquadro 2 l'immagine del buffer che si svuota, mentre nel riquadro 5 compare la scritta:

    WRITING DATA FROM BUFFER

12. Ripetere le manovre da 9 a 12 finché non è stato copiato tutto il disco originale. Compare allora nel riquadro 5 la scritta:

    BACKUP FINISHED

    e nel riquadro 6 la scritta:

    REMOVE DESTINATION DISK FROM DRIVE

    La copia è terminata. Si può estrarre il nuovo disco dal drive.

COPIA CON DUE DRIVE

Per eseguire la copia di un disco usando due drive occorrono i due programmi "Disk Addr Change" e "Copy/all", entrambi contenuti del disco "1541 Test/Demo" che accompagna ogni drive Commodore 1541.
Prima di iniziare la copia, occorre cambiare da 8 a 9 il codice di indirizzo del secondo drive. Se si dispone in permanenza di un sistema Commodore 64 con due drive, conviene rendere permanente tale assegnazione (ricorrendo magari alla consulenza del rivenditore Commodore).
Il cambio del codice di indirizzo è eseguito dal programma "Disk Addr Change", la copia vera e propria dal programma "Copy/all".

Il computer Commodore considera normalmente per ogni drive il codice di indirizzo 8, a meno che non venga modificato esplicitamente. L'assegnazione di un nuovo codice al secondo drive è necessaria perché non è possibile eseguire una copia da un indirizzo ad un altro con lo stesso codice.
Ecco le manovre da seguire:

1. Accendere il primo drive e inserirvi il disco "1541 Test/Demo", battendo poi:

   LOAD "DISK ADDR CHANGE",8

   Il computer scrive:

   SEARCHING FOR DISK ADDR CHANGE
   LOADING
   READY

2. Bisogna poi seguire le istruzioni che compaiono sullo schermo. Ecco la prima:

   DRIVE ADDRESS CHANGE PROGRAM
   TURN OFF ALL DRIVES NOW
   EXCEPT THE ONE TO BE CHANGED
   OLD DEVICE ADDRESS? 8
   traduzione:
   programma di modifica codice di indirizzo
   spegnere tutti i drive
   tranne quello che deve cambiare codice
   qual è il vecchio codice? 8

   Si spegne allora il primo drive (quello che non cambia codice) e si accende il secondo (quello che deve cambiar codice); poi si batte il tasto RETURN.
3. Compare sullo schermo la scritta:

   NEW DEVICE ADDRESS? 9

   Si batte il tasto RETURN.
4. Compare la nuova scritta:

   THE SELECTED DRIVE HAS BEEN CHANGED
   NOW TURN ON THE OTHER DRIVE(S)
   READY

Si riaccende allora il primo drive: tutto è pronto per la copia. Ecco le manovre per eseguire la copia, che avviene dal drive che ha codice 8 al drive che ha codice 9 (entrambi accesi):

1. Con il disco "1541 Test/Demo" inserito nel drive 8, battere:

   LOAD "COPY/ALL",8   (e tasto RETURN)

2. Il computer scrive:

   SEARCHING FOR COPY/ALL
   LOADING
   READY

3. Quando compare la scritta READY, battere:

   RUN   (e RETURN).

4. Compare ora sullo schermo una serie di messaggi (bisogna battere per quattro volte il tasto RETURN):

   DISK.COPY.ALL   JIM BUTTERFIELD
   FROM UNIT? 8
   DRIVE 0
   TO UNIT? 9
   DRIVE 0

5. A questo punto, compare la scritta:

   WANT TO NEW THE OUTPUT DISK
   ? N

   Si estrae allora dal drive 8 il disco "1541 Test/Demo", sostituendolo con l'originale che va duplicato, e si inserisce nel drive 9 il nuovo disco.

6. Supponendo di avere inserito nel drive 9 un disco nuovo, lo si deve inizializzare, perciò si batte:

   Y

7. Il computer scrive:

   DISK NAME, ID?

Bisogna battere un nome (lungo fino a 16 caratteri), seguito da un numero con due cifre qualunque, ad esempio:

DISCOMAURO, 09

Si aspetta un paio di minuti per completare l'inizializzazione del disco nel drive 9.

8. Quando il nuovo disco è inizializzato, compare la scritta:

   PATTERN?*

   In questo modo, il programma di servizio domanda se si vuole copiare l'intero disco. Si risponde battendo il tasto RETURN.
9. A questo punto inizia la copia. Bisogna solo aspettare... Man mano compare l'elenco dei file presenti sul nuovo disco (il file in corso di copia è indicato in fondo).
10. Quando la copia è terminata, si può estrarre il disco. Se si vuole, si può proseguire con un'altra copia.

B.3 Come ricopiare file da un disco all'altro

Si può tenere copia dei file più importanti registrandoli ogni volta su due dischi, quello che si usa di solito e uno di riserva (che deve essere però già inizializzato). Se si fa così, dopo aver registrato il file che interessa nel modo solito, si inserisce nel drive il nuovo disco e si ripete su questo la registrazione; poi si estrae il disco di riserva, si inserisce il disco di servizio normale e si prosegue. Si può anche eseguire la copia di un file in un altro momento, riprendendolo dal disco in cui è registrato. Supponiamo per esempio di voler ricopiare su un secondo disco il file chiamato "NAVI.
Si ripulisce la memoria del computer, battendo:

CIAO

oppure:

CANCELLA TUTTO

Poi si inserisce nel drive il disco che contiene il file "NAVI e si batte:

RECUPERA "NAVI

Si sostituisce ora nel drive il primo disco con il disco nuovo e si batte:

CONSERVA "NAVI

B.4 Come ricopiare parti di file

Supponiamo ora di voler ricopiare solo alcune delle procedure di un file. Si ripulisce per prima cosa la memoria del computer, battendo come di consueto CIAO oppure CANCELLA TUTTO.
Si carica poi in memoria il file che interessa, con: RECUPERA "NAVI.
Naturalmente, al posto di "NAVI si scriverà il nome del file che di volta in volta interessa.
Si inserisce quindi nel drive il secondo disco e si batte:

STP

Si ottengono così i titoli delle procedure contenute nel file e ora in memoria. Si possono quindi cancellare, una alla volta, le procedure che non vanno ricopiate, ad esempio:

CANCELLA RETTANGOLO1
CANCELLA VELA1
CANCELLA ...

Si può anche cancellare un'intera lista di procedure:

CANCELLA [RETTANGOLO1 VELA1 VELA2]

Alla fine, si batte ancora:

STP

per vedere se sono rimaste in memoria proprio le procedure che si vogliono ricopiare. Poi si registrano sul nuovo disco.
Se si vogliono aggiungere le procedure ad un file già esistente, invece di crearne uno nuovo, si carica in memoria il file che interessa, prima di registrare il tutto. Supponiamo ad esempio che il file già esistente si chiami "AUTOMOBILI. Si batte:

RECUPERA "AUTOMOBILI

Poi si controlla il contenuto della memoria:

STP

A questo punto, ci dovrebbero essere in memoria tutte le procedure di "AUTOMOBILI e quelle già prese da "NAVI. Si registrano allora tutte insieme:

CONSERVA "AUTOMOBILI

Oppure, se vanno registrate solo alcune di queste procedure, si batte:

(RECUPERA "AUTOMOBILI [FERRARI FIAT MASERATI VELA1 VELA2])

Come si vede, basta indicare il nome del file e poi la lista dei nomi delle procedure in questione (quando il comando RECUPERA ha più di due argomenti, va posto fra parentesi con tutti i suoi argomenti).

B.5 Come aggiornare i file

Quando si prepara un nuovo programma, o magari si inseriscono molti dati in un programma già pronto, conviene registrare di tanto in tanto su disco il lavoro fatto, per non correre il rischio di perderlo (per un nostro errore, per un'interruzione dell'alimentazione elettrica, ecc.). Invece di usare sempre il solito file, ad esempio "NAVI, si possono impiegare diversi file, ciascuno con una versione diversa degli stessi dati. Ad esempio, si incomincia con:

CONSERVA "NAVI1

Dopo un po', si salvano i nuovi dati con:

CONSERVA "NAVI2

Dopo un altro po', si registra quanto si è fatto con:

CONSERVA "NAVI3

e così via.
Se si batte il comando CATALOGO, si vede che il disco contiene molti file, cioè tutte le diverse versioni man mano registrate. Si possono cancellare quelle che non interessano più, con il comando CANCEARC, liberando così spazio su disco.

B.6 Come tenere i dischi in buono stato

Ecco qui alcune regole, da tenere sempre in mente per evitare brutte sorprese.

BISOGNA:

- Conservare i dischi nelle loro bustine.
- Tenere i dischi in una scatola che li ripari dalla polvere, dagli urti e... dalla confusione.
- Segnare su ogni disco il suo contenuto, usando le apposite etichette (si scrive con pennarelli a feltro morbido, non con matite o penne a biro, che possono danneggiare il disco). Anche qui, il pericolo numero uno è ... la confusione (provate ad avere anche solo sette o otto dischi, e vedrete che ci si confonde presto).

NON BISOGNA:

- Lasciare i dischi in giro, magari senza bustina.
- Appoggiare i dischi su caloriferi, stufette, ecc. oppure lasciarli esposti al sole.
- Lasciare i dischi vicino a un corpo magnetizzato o a un motore elettrico.
- Scrivere sui dischi con matite o penne a sfera.
- Maltrattare i dischi, facendoli cadere o strapazzandoli in vario modo.

Elenco dei comandi Logo usati in questo libro

Sono raggruppati qui tutti i comandi Logo usati nel libro, raggruppati per argomento. Se c'è, viene anche indicata l'abbreviazione. Compaiono anche esempi di applicazione dei comandi con argomenti.

C.1 Manovra della tartaruga

Nome	Abbreviazione	Esempi
DISEGNA		
ESCIDIS	ED	
AVANTI	A	AVANTI 20, A 20
INDIETRO	I	INDIETRO 10, I 10
DESTRA	D	DESTRA 30, D 30
SINISTRA	S	SINISTRA 90, S 90
SULAPENNA	SU	
GIULAPENNA	GIU	
NASTARTA	NT	
MOSTARTA	MT	
ATTORNO		
MURO		
DUECOLORI		
UNCOLORE		
ASCOL		ASCOL 3
COLORESFONDO	CS	COLORESFONDO 4, CS 4
PULISCISCHERMO	PS	
TANA		
GRASCHERMO	tasto f5	

Nome	Abbreviazione	Esempi
DISCHERMO	tasto f3	
TESCHERMO	tasto f1	
VAXY		VAXY 100 30, VAXZY 10 (−20)
VAX		VAX 60, VAX (−100)
VAY		VAY 35, VAY (−20)
ASDIR		ASDIR 90
XCOR		STAMPA XCOR, VAX XCOR+20
YCOR		STAMPA YCOR, VAY YCOR−20
DIREZIONE	DIR	SE DIREZIONE = 0 STOP STAMPA DIREZIONE
STATOSCHERMO	SS	STAMPA STATOSCHERMO

C.2 Composizione di testo e gestione di file

Nome	Abbreviazione	Esempi
PER	END	PER RETTANGOLO
FINE		
EDITA	E	EDITA RETTANGOLO, E RETTANGOLO
FAIVEDERE	FV	FAIVEDERE RETTANGOLO, FV NOMI
STP		
CANCELLA	CA	CANCELLA RETTANGOLO, CA TUTTO
CONSERVA		CONSERVA "CERCHI
RECUPERA		RECUPERA "CERCHI
CATALOGO		
CANCEARC		CANCEARS "ANTICAGLIA
CONSERDIS		CONSERDIS "CAMION
RECUPDIS		RECUPDIS "CAMION
CANCEDIS		CANCEDIS "CAMION

C.3 Input, output e stampa

Nome	Abbreviazione	Esempi
STAMPA	ST	STAMPA [CARO MIO]
SCRIVI		SCRIVI [INDOVINA UN NUMERO]
PULISCITESTO	PT	
LEGGIRIG	LR	ASSEGNA "RISPOSTA LEGGIRIG
TASTO?		SE TASTO? RIPORTA LEGGIRIG
LEGGICAR	LC	ASSEGNA "TASTO LEGGICAR
STAMPANTE		
NOSTAMPANTE		

C.4 Operazioni su numeri

Nome	Abbreviazione	Esempi
+		AVANTI :GRANDEZZA+10, STAMPA 5+3
–		STAMPA 5–3, AVANTI :GR–10
*		STAMPA 3*5, AVANTI :GR*4
/		STAMPA 360/3, DESTRA 360/3
RADQ		STAMPA RADQ 25
ACASO		STAMPA ACASO 20
SEME		
NUMERO?		SE (NON NUMERO? :RISPOS) RIPORTA LEGGINUMERO

C.5 Operazioni su parole, liste e variabili

Nome	Abbreviazione	Esempi
ASSEGNA	A	ASSEGNA "PART DIREZIONE ASSEGNA "GRAN 50
COSA		STAMPA COSA :DIS
PAROLA		STAMPA PAROLA "a "A
FRASE	FR	STAMPA FRASE [CIAO] [CARO] STAMPA FRASE "CIAO [CARO] STAMPA (FR [CIAO] [CARO] [MIO]
PRIMO		STAMPA PRIMO "CIAO STAMPA PRIMO [CIAO AMICO MIO]
MENPRI	MP	STAMPA MENPRI "CIAO STAMPA MP [CIAO CARO MIO]
ULTIMO		STAMPA ULTIMO "CIAO STAMPA ULTIMO [CIAO CARO MIO]
MENULT	MU	STAMPA MU "CIAO STAMPA MENULT [CIAO CARO MIO]
CONTA		STAMPA CONTA [1 2 3 4]

C.6 Controllo delle procedure e comandi condizionali

Nome	Abbreviazione	Esempi
RIPETI		RIPETI 4[AVANTI 20 DESTRA 90]
SE		SE :GRAN< 10 STOP
STOP		SE :GRAN> 30 STOP
<		SE :GRAN< 30 STOP
>		SE :GRAN> 80 STOP
=		SE :GRAN = 110 STOP
VERIFICA		VERIFICA :RISPOSTA = 7
SEVERO	SEV	SEVERO STAMPA [BENE]
SEFALSO	SEF	SEFALSO STAMPA [NO!]
TUTTIVERI?		SE TUTTIVERI? (YCOR> 0) (XCOR< 7) STOP
UNOVERO?		SE UNOVERO? (XCOR> 0) (YCOR< 6) STOP
NON		SE NON (XCOR> 0) STOP
		SE (NON NUMERO? :RISPOS) RIPORTA LEGGINUMERO
PUNTOACAPO		SE :COM = "E PUNTOACAPO
RIPORTA	RI	RIPORTA "FALSO, RI LUNGH + 10
TRACCIA		
NOTRACCIA		

C.7 Controllo del funzionamento diretto del Logo

Tasto	Effetto
RETURN	Invia al Logo il comando battuto alla tastiera
INST/DEL	Cancella il carattere a destra del cursore
←	Sposta il cursore di un carattere verso sinistra
SHIFT ⇐CRSR⇒	Sposta il cursore di un carattere verso sinistra (con autoripetizione, se si tengono premuti)
⇐CRSR⇒	Sposta il cursore di un carattere verso destra (con autoripetizione, se si tiene premuto)
CTRL-G	Arresta il Logo
f5	Dedica alla grafica lo schermo intero
f3	Divide lo schermo fra grafica e testo
f1	Dedica lo schermo intero al testo

C.8 Controllo dell'editor del Logo

a) *Spostamento del cursore*

Tasto	Effetto: il cursore si sposta ...
←	... di un carattere verso sinistra
SHIFT ⇐CRSR⇒	... di un carattere verso sinistra (con autoripetizione, se li si tiene premuti)
⇐CRSR⇒	... di un carattere verso destra (con autoripetizione, se è tenuto premuto)
SHIFT ⇑ CSRS⇓	... di una riga verso l'alto
⇑CRSR⇓	... di una riga verso il basso (con autoripetizione, se è tenuto premuto)
CTRL-L	... alla fine della riga
CTRL-A	... all'inizio della riga
CTRL-F	... in basso, per uno schermo intero
CTRL-B	... in alto, per uno schermo intero

b) *Composizione di testo*

Tasto	Effetto
INST/DEL	Cancella il carattere alla sinistra del cursore
CTRL-D	Cancella il carattere su cui si trova il cursore
CTRL-K	Cancella la parte di riga alla destra del cursore
RETURN	Sposta in giù il cursore di una riga (trascinando il testo alla sua destra)
CTRL-O	Apre una nuova riga al cursore (tutto il testo alla sua destra si sposta in giù)

c) *Termine della composizione di testo*

Tasto	Effetto: richiama il funzionamento diretto del Logo e...
CTRL-C RUN/STOP	... definisce le procedure appena composte, assegnandole al Logo
CTRL-G	... annulla il testo appena composto

Indice analitico

La McGraw-Hill pubblica in tutto il mondo centinaia di libri di informatica per lo studio, la professione e il tempo libero. La produzione in lingua italiana comprende:

88 7700 001 5	J. Heilborn e R. Talbott, *Guida al Commodore 64*
88 7700 002 3	C.A. Street, *La gestione delle informazioni con lo ZX Spectrum*
88 7700 003 1	T. Woods, *L'Assembler per lo ZX Spectrum*
88 7700 004 X	R. Jeffries, G. Fisher e B. Sawyer, *Divertirsi giocando con il Commodore 64*
88 7700 005 8	G. Bishop, *Progetti hardware con lo ZX Spectrum*
88 7700 006 6	H. Mullish e D. Kruger, *Il BASIC Applesoft*
88 7700 007 4	N. Williams, *Progettazione di giochi d'avventura con lo ZX Spectrum*
88 386 0008 2	H. Peckham, *Il BASIC e il PC-IBM in pratica*
88 7700 009 0	H. Peckham, *Il BASIC e il Commodore 64 in pratica*
88 7700 010 4	S. Nicholls, *Tecniche avanzate in Assembler con lo ZX Spectrum*
88 7700 011 2	K. Skier, *L'Assembler per il Commodore 64 e il VIC-20*
88 7700 012 0	S. Kamins e M. Waite, *Programmazione umanizzata in Applesoft*
88 7700 013 9	A. Pennell, *Guida allo ZX Microdrive e all'Interface 1*
88 7700 015 5	P. Cohen, *Grafica e animazione con gli Apple II*
88 7700 016 3	C. Duff, *Guida al Macintosh*
88 7700 017 1	G. Kane, *Il manuale MC68000*
88 386 0018 X	P. Hoffman e T. Nicoloff, *Il manuale MS-DOS*
88 386 0019 8	E.M. Baras, *Come usare il Symphony*
88 7700 020 1	S. Nicholls, *Grafica avanzata con lo ZX Spectrum*
88 7700 021 X	L.J. Graham e T. Field, *Guida al PC-IBM*
88 7700 022 8	T. Field, *Come usare MacWrite e MacPaint*
88 7700 024 4	H. Peckham, *Il BASIC e gli Apple II in pratica*
88 7700 025 2	C. Morgan e M. Waite, *Il manuale 8086/8088*
88 7700 026 0	W. Ettlin, *Come usare il Multiplan*
88 7700 027 9	G. Mainis, *Il manuale ProDOS*
88 7700 028 7	J. Jones, *Il SuperBASIC del QL*
88 7700 029 5	C. Opie, *L'Assembler per il QL*
88 7700 030 9	W. Ettlin e G. Solberg, *Il BASIC Microsoft*
88 386 0031 7	D.L. Toppen, *Il Forth in pratica*

88 7700 032 5	R. Person, *Le meraviglie dell'animazione con gli Apple II*
88 7700 034 1	P. Hoffman, *Il manuale MSX*
88 7700 035 X	R. Person, *Le meraviglie dell'animazione con il PC-IBM*
88 7700 036 8	W. Ettlin, *Il Multiplan per il Macintosh*
88 386 0037 6	D. Kruglinski, *Introduzione al Framework*
88 386 0038 4	L. Barnes, *Come usare il dBase II*
88 386 0042 2	D. Watt, *Il LOGO per il Commodore 64*
88 386 0044 9	D. Kater e R. Kater, *Guida alle stampanti Epson*
88 7700 601 3	S. Harrington, *Computer Graphics - Corso di programmazione*
88 7700 602 1	O. Lecarme e J.L. Nebut, *Pascal - Guida per programmatori*
88 386 0603 X	M. McGilton e R. Morgan, *Il sistema operativo UNIX*

In preparazione

88 386 0033 3	P.A. Sand, *Programmazione avanzata in Pascal*
88 386 0040 6	L. Barnes, *Come usare il dBase III*
88 386 0043 0	T.J. Byers, *Guida al PC-AT IBM*
88 386 0045 7	R.L. Tokheim, *Progetti di circuiti elettronici per microcalcolatori*

Il Logo, linguaggio sviluppato al MIT e destinato alla prima alfabetizzazione informatica, si è imposto non solo come strumento per stimolare i bambini a familiarizzare con il computer ma anche come "ambiente" educativo globale.
Il libro introduce le semplicissime regole di questo linguaggio attraverso numerose attività, facilmente realizzabili anche dai più giovani lettori, da soli o con l'aiuto di genitori ed educatori, applicate al Commodore 64, attualmente uno dei più diffusi home computer.
Il Logo per il Commodore 64 può essere utilizzato come manuale per l'insegnamento familiare o scolastico e contiene, suddivisa nei vari capitoli, una guida preziosa per chi accompagna i bambini nella loro scoperta del computer e del mondo della logica.
Ogni aspetto che rappresenta una potenziale difficoltà viene evidenziato per aiutare i lettori a muoversi in questo linguaggio: l'impostazione volutamente elementare dei comandi Logo, infatti, non deve trarre in inganno, perché questo approccio estremamente amichevole nasconde la possibilità di sviluppare programmi che non hanno nulla da invidiare a quelli realizzati con linguaggi ritenuti più potenti.

Un libro di BYTE

Lire 32 000
(IVA 2% inclusa)

ISBN 88 386 0042 2

D. Watt

Il Logo per il
Commodore 64

McGraw-Hill